POLITECNICO DI TORINO

MECHANICAL AND AEROSPACE ENGINEERING DEPARTMENT

Master degree program in Aerospace Engineering

# Design of Observer-based Navigation Algorithms for a CubeSat Space Mission

**Supervisors**
prof. Elisa Capello
prof. Hyeongjun Park

**Candidate**
Davide Neglia

Academic year 2020-2021

...

*Ai miei genitori, i miei fratelli e mio nonno Mimino*

# Ringraziamenti

Grazie mamma e papà per avermi sempre spronato ad essere migliore e a fare del mio meglio. Siete i miei punti di riferimento, il mio posto sicuro e il mio esempio. Mamma, sei una donna speciale e farò in modo di renderti fiero di me, qualcunque sia il mio percorso futuro.

Grazie Francesco e Domenico, i due fari che hanno da sempre illuminato la mia strada. La costante voglia di emularvi mi ha reso la persona che sono adesso, e questo risultato ve lo devo. Grazie a voi mi sono sempre sentito protetto e mai solo, qualsiasi fosse la sfida da affrontare. Diventeremo uomini assieme, non vedo l'ora.

Grazie a mio nonno Mimino, mi hai insegnato cosa sia l'amore, la dedizione e la passione per il lavoro. Sono fortunato ad essere tuo nipote e porterò sempre con me i tuoi insegnamenti. Anche se non tornerò presto giù da te sappi che ti ho sempre nel cuore, e prima o poi tornerò in patria.

Grazie a zio Martino, zia Cinzia e Chiara, una seconda famiglia. Grazie per il bene che mi volete e per esser stati sempre pronti ad aiutarmi.

Grazie a tutto il resto della mia famiglia, per esserci sempre stati e per il calore che mi mostrate ogni volta che ci vediamo.

Grazie ai miei compagni di viaggio di questa fantastica avventura Luca, Simone, Luca, Greta, Ale e Zaba. Senza di voi nessun esame sarebbe stato piacevole da studiare. Grazie per la vostra costante presenza, per il vostro aiuto e per le risate che hanno sempre accompagnato il nostro stare insieme. La vita ci riserverà il meglio, saremo uomini e donne migliori assieme ancora dopo questo percorso ormai giunto al termine.

Grazie a Paolo, Antonella, Claudio, Camilla e Stefano. Se non ci foste stati nella mia vita, non sarei la persona che oggi tutti conoscono. Grazie per la vostra continua mano tesa. Siamo e saremo un team, a prescindere da cosa ci riserverà

il futuro. Grazie per l'amore incondizionato, grazie per avermi insegnato sempre tutto quello che sapevate. Grazie all'esperienza con voi ho realizzato che è possibile trasformare una passione in un lavoro e vi prometto che farò dei miei sogni realtà e che contribuirò finchè potrò alla crescita di questo ambiente fantastico.

Grazie a tutti i Dreamer di Memory Up e ai miei tutorati, per essere fonte di stimolo continuo. Essere un esempio per voi è un grande onore, vi prometto che continuerò a migliorarmi per continuare a inspirarvi a raggiungere i vostri sogni.

Grazie a tutte le persone che ho incrociato nel mio cammino, che hanno reso la mia esperienza qui a Torino indimenticabile.

Grazie alla professoressa Capello, per la costante presenza e pazienza, e per i consigli che vanno ben oltre l'ambito universitario. Thank you Professor Park, although I wasn't able to be in New Mexico, I really enjoyed your advice and I feel grateful to have met you.

Grazie a Giulia, senza il quale la grammatica inglese di questa tesi sarebbe parecchio scorretta. Grazie per esserci sempre, per amarmi e per prenderti cura di me. Sono fortunato ad averti nella mia vita e sono felice di averti accanto ora che mi affaccio in una nuova fase della mia vita. Sei preziosa ed unica e farò in modo che la tua unicità la conosca il mondo intero, ti sarò sempre a fianco e farò in modo che i nostri sogni si realizzino il prima possibile. Grazie per esserci, per supportarmi e per sopportarmi. Ti amo.

Infine grazie alla persona che sono diventato in questi anni, che il futuro mi riservi grandi cose e che la vita mi continui a regalare la felicità che mi ha sempre donato.

**Abstract**

Nowadays the design and operation of GNC of small satellites is widely being studied, due to the current and past space missions designed by several industries, research institutes, and space agencies. Some in-orbit demonstrations require heavy and bulky payload that must be embedded with complex set of sensors, actuators, and maintenance systems. Thus, the redundancy of elements is always inferior, so that a mathematical model estimating variables is needed to achieve the same goal with less sensors. At the same time, a good control of the state variables can be helpful in order to reduce fuel consumption, with the consequence of less space occupied by the actuation system or in an increase of the operational life span of the satellite. The aim of this thesis is to design an orbital simulator of a rendezvous manoeuvre of a 6U CubeSat. In particular, the main purpose is to design different observer-based navigation algorithms, to estimate all the system variables involved in the mission. At the same time, the system should be controlled to reach the desired values of attitude, position, and velocity.

In this thesis, different state estimation methods are investigated, considering different Sliding Mode Observers (SMOs), and including a comparison with classical Kalman filters. The performance indices considered for the comparison are the estimation error, convergence time, and maintenance of performance in presence of disturbances. This thesis presents the results that can be obtained by applying the estimation techniques and the main disadvantages encountered during the development of the simulation model. Despite of the presence of external disturbances, process noise and measurements noise the performance of the studied filters always maintain good levels. A comparison of the proposed different techniques is performed for translational and rotational motions of the CubeSat. For example, the $1^{st}$ order Sliding Mode Observer shows the best results in the estimation of the position along the y axis. On the other hand, as regard as the estimation of the position along the x and z axis, that have a coupled dynamic, the best result is given by the Kalman Filter and the $2^{nd}$ order Sliding Mode Observer, in particular by the Super Twisting Observer. The Kalman filter and the Extended Kalman filter also shows the best results for the estimation of the Angular Velocities and the quaternion vector.

Different control systems have been analyzed with the aim of identifying an optimal algorithm for the control of the attitude and the position. The first objective is to design a controller for the position control. Some controllers are studied, but only two of these are applied in the simulation model: the Sliding Mode Control and the PID controller. The second objective is to find a suitable controller for the

attitude. The first model implemented is a Quaternion Feedback Controller (QFC), which is used as comparison with the second implemented technique, based on Linear Quadratic Gaussian (LQG) method. This consists of a Kalman Filter combined with a Linear Quadratic Regulator (LQR). Moreover, the Linear Quadratic Controller combined with different state observers is studied and applied in simulations and compared with the result given by the classical LQG controller.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Nowadays the space environment is chosen by industries, research institutes and space agency as new place in which can be executed advanced experiments. More and more a new space race is being experienced to develop new technologies used for different applications. One of the new challenges is to reduce the size of the satellites in order to reduce costs, in fact, a smaller satellite is light, and this means less launch and maintenance costs. Another challenge is to increase the internal space of the satellite to obtain more space for payload accommodation. One way to achieve this goal is to reduce the number of components needed by the Guidance Navigation and Control system, like the sensors, maintaining the same performances.

In the first part of this chapter, four state observers are studied in order to estimate the state variable in translational and rotational motion. The main goal is to reduce as much as possible the estimation error and evaluate the observer performance for each dynamic system involved in the simulations despite of the presence of disturbances. These disturbances come from the external environment or can be the signal and process noise. Both the disturbances are included in the mathematical model and considered into the simulations. The State Observers chosen and compared are classical Kalman Filter, the Extended Kalman Filter, the 1$^{st}$ order Sliding Mode observer, and the Super-Twisting Observer. For the attitude variable estimation, the initial condition of the system are varied to test the effectiveness of the estimate under highly unstable conditions. The result of this work is shown in Chapter 5.

Another way to increase the internal space of a satellite is to reduce the space occupied by the actuation system. To do this, more and more attention is given to the control algorithms that can ensure the control of the state variables that allow to save more fuel, with the consequence of reduce tanks volumes and increase space for payload. In this chapter, the state of art of control algorithms has been analysed with the objective to evaluate the performance of control algorithms for each variable to maintain the desired attitude, position, and velocity. Two control algorithms have been studied and designed for the simulations. For the attitude control, the controller chosen is the Linear Quadratic Gaussian (LQG), compared with the classical Quaternion Feedback Controller while for the position control, the PID controller was selected. The mission taken into account in this thesis work is a rendezvous and docking manoeuvre performed by 6U CubeSats.

## 1.1 State Observer

In control theory, a state observer, or state estimator, is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. This property is necessary to solve many control theory problems. For example, in most practical cases, the physical state of the system cannot be determined by direct observation and the system can be stabilized by using a state feedback loop. Instead, indirect effects of the internal state are observed by the system outputs. If a system is observable, it is possible to fully reconstruct the system state from its output measurements using the state observer.

In this section, four state observers are introduced and detailed in Chapter 3 to individuate the performance of the state estimators for the state variables involved in the simulations.

#### Kalman Filter

The Kalman Filter (KF) allows the estimation of certain variables. It can be used to estimate a state of system by combining measurements from different sources that can be subjects by noise. This filter can be used to find the best combination between the properties of various sensors, estimating the exact value of a certain parameter [10].

The KF is used when:

- The variables of interest can be only measured indirectly;
- Measurements are available from various sensors but might be subject of noise;

The Kalman Filter combines the measurement and the prediction to find the optimal estimate of a certain value. In fact, if there are multiple state variables that are measured and estimated by the mathematical model, those states are compared step by step with the measurements by the Kalman Filter by multiplying the prediction and the measurement together, scaling the result and computing the mean of the resulting probability density function.
The Kalman Filter is a two steps process: the first part predicts the current state by using state estimate from the previous time step and current input. It represents an a priori estimate since is calculated before the current measurement. The

second part takes measurement and incorporate it into the prediction to update the a priori estimate with the result of a posteriori estimate [11]. The Kalman Filter is also referred to a sensor fusion algorithm. If we have multiple sensor measurements, the dimensions of the state vector would change; however but the main logic of the filer will remain the same [12].

### Extended Kalman Filter

If a nonlinear function is taken into account, which doesn't follow a Gaussian distribution, the Kalman filter is not applicable anymore. To solve this problem, a *local linearization* can be done. The local linearization happens thanks to the first derivative of the prediction and correction function, the *Jacobian matrices*.

The EKF linearizes the nonlinear function around the mean of the current state estimate. At each time step, the linearization is performed locally and the resulting Jacobian matrices are then used in the prediction and update states of the Kalman Filter algorithm [14].

The main drawbacks are:

- It is difficult to calculate the Jacobians if they need to be found analytically;
- There is a high computational cost if the Jacobians are found numerically;
- It cannot be applied an EKF to systems with a discontinuous model, since the system is not differentiable and the Jacobian wouldn't exist;
- Linearization doesn't provide a good approximation for highly nonlinear systems. Linearization becomes invalid since the nonlinear function cannot be approximated well enough by a linear function and doesn't describe system dynamic.

However, to address the issues with EKFs, other estimation techniques can be used: the *Unscented Kalman Filter* (UKF) and the *Particle Filter*. In the first one, the filter approximates the probability distribution. This one selects a minimal set of sample points such that their mean and covariance is the same as distribution. These are called $\sigma$ point. Each $\sigma$ point is then propagated through the nonlinear system model. The points are calculated and then an empirical Gaussian distribution is computed, which is used to calculate the new state estimate [9].

As concerning the *Particle Filter* (PF), it is good to say that it differenciate from the Unscented Kalman Filter since the PF approximates any arbitrary distribution. To make this approximation the number of particles that a particle filter needs is much larger than needed by an UKF [8]. For nonlinear systems, EKF, UKF or PF can be applied. As regard the computational cost, the Particle Filter is computationally the most expensive filter since it requires a large number of particles to approximate the distribution. The EKF will be used to estimate the positions and the velocities USING the measurements given by the Inertial Navigation Unit (IMU).

### 1$^{st}$ order Sliding Mode Observer

Sliding mode observers based on first-order sliding modes are effective in the presence of uncertainties or disturbances. When the relative degree of the outputs is one, with respect to the uncertainties or disturbances, and when there is not needed differentiation of noisy output, this observer can be applicable.

For any observation of a mechanical system, in which we have measured position, the estimation of velocity is necessary. The uncertainties/disturbances in mechanical systems are in the equations for accelerations and have relative degree two with respect to the measured positions. This means that differentiators, which can provide the best possible accuracy in the presence of sampling steps and noise, are needed for the general case of observation of control systems working under uncertainties/disturbances [1].

The ability to generate a sliding motion on the error between the measured plant output and the output of the observer ensures that a sliding mode observer produces a set of state estimates that are precisely commensurate with the actual output of the plant. Analysis of the value of the applied observer injection signal, the so-called equivalent injection signal, contains useful information about the mismatch between the model used to define the observer and the actual plant. The discontinuous injection signals, which are perceived as problematic for many control application, have no disadvantages for software based observer framework [13].

The results given in Section 5.2 will show these particular properties of a 1$^{st}$ order Sliding Mode Observer. Nevertheless, it will be shown that despite of a great estimation of single state variable, the simulation of the estimation of multiple state system will not give a good result.

### Super-Twisting Sliding Mode Observer

One of the popular second-order sliding mode algorithms offering a finite reaching time and useful for sliding mode based observation is the Super-Twisting Algorithm considered in Chapter 3. The sliding mode approach has been exceptionally successful in the design of state feedback controllers. By the way, in the great majority of the physical systems, an output is available for measurement. In that case, other states of the system can be obtained using an observer. The Sliding Mode Observers are widely used due to the finite time convergence and for the estimation of the uncertainty [20].

The dynamical system of the Super-Twisting Observer (STO) includes disturbances and correction terms. Once defined the error variable as $e_1 = x_1 - \hat{x_1}$ and $e_2 = x_2 - \hat{x_2}$, the correction terms are selected and it can be defined the error dynamic as described inChalanga et al. [20].

If the parameters that define $\dot{e_1}$ and $\dot{e_2}$ are correctly chosen, it may be noticed how the error $e_1$ and $e_2$ will go to zero simultaneously in a finite time, thus representing the best advantages of the STO.

This observer will be applied into the observation on the position along the x and z axis. The coupled dynamics is completely observed with good results. The error between the real value and the estimated one reaches zero in a finite time and maintains this value until the end of the maneuver. In Section 5.2 will be exhibited the entire mathematical model and the application into the orbital simulator.

## 1.2 Control systems

In the formulation of any practical control problem, there will be always a discrepancy between the actual plant and its mathematical model used for the controller design. These discrepancies (or mismatches) arise from unknown external disturbances, plant parameters, and parasitic/unmodeled dynamics. Designing control laws that provide the desired performance to the closed-loop system in the presence of these disturbances/uncertainties is a challenging task for a control engineer.

In this section, several control algorithms are showed to identify the one chosen for each variable in order to control the attitude and the position. The Sliding Mode Control, the PID control, the Quaternion Feedback Controller and the Linear Quadratic Gaussian are here introduced and studied in Chapter 4. Moreover, the state of art of different adaptive controllers is shown in the final part of this section.

#### Sliding Mode Control

One particular approach to robust controller design is the so-called *Sliding Mode Control* technique [1], which is in the class of a nonlinear control technique. The basic idea is to design a sliding surface described by *s=0* on which the sliding mode along this surface begins after the finite time when the trajectories of the system have reached the surface. The idea is to make the system slide on a plane and the goal is that the system goes on that plane, slides on this and goes to the origin that always represents the point of stability. To do this there are two steps:

- Define the sliding surface that represents a kind of degree of freedom. By defining the sliding surface in a proper way, is being defined what should be the behavior of our system.
- Once this is done, a feedback law of the states must be defined to bring the trajectories of the plant to be controlled precisely on this surface.

In Section 4.1.2, the mathematical model of this controller will be deepened with some examples regarding this type of controller, the positive aspects, and the negative aspects.

### PID Control

The main objective is to control both attitude and position variables. The first control method studied and applied on this project is the *PID controller*, used to control the position along the y-axis, and to control x and z position during the final phase of the entire maneuver, the cone of approach.

The PID is a continuous controller used, for example, in the early 20$^{\text{th}}$ century to control ship steering for US Navy. This was then given a mathematical treatment by the Russian American engineer Nicolas Minorsky. The main goal was the stability, not general control, which simplified the problem significantly.

In general it can be said that the use of the PID algorithm does not guarantee optimal control of the system or its control stability. The fundamental difficulty with PID control is that it is a feedback control system, with constant parameters, and no direct knowledge of the process, and thus overall performance is reactive and a compromise. In addition, situations may occur where there are excessive delays: the measurement of the process value is delayed, or the control action does not apply quickly enough. The response of the controller can be described in terms of its responsiveness to an error, the degree to which the system overshoots a set-point, and the degree of any system oscillation. But the PID controller is broadly applicable since it relies only on the response of the measured process variable, not on knowledge or a model of the underlying process. In Section 4.1.1, is discussed the mathematical model and a simple example of PID control loop with the results given by the simulation.

### Quaternion Feedback Controller

Quaternions can be easily computed by modern Attitude Determination and Control System (ADCS) and for this reason their use is very common. As a consequence a simple feedback control law based on the information obtained from the attitude sensors is easily implementable to obtain an autonomous maneuver.

Thanks to the use of the quaternion error a feedback law is globally asymptotically stabilizing onto any arbitrary desired attitude, $(q_{des0}, \mathbf{q_{des0}})^T$, for a wide choice of the gain $K_p$ and $K_d$ [38]. This controller will be used in order to compare the results with the Linear Quadratic Gaussian controller.

#### Linear Quadratic Gaussian

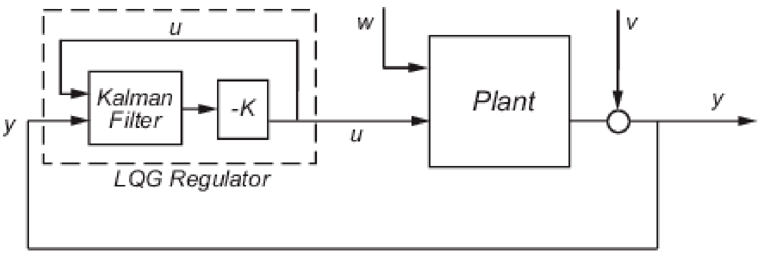


Figure 1.1. Linear Quadratic Gaussian [34].

As shown in Figure 1.1, there is a schematic illustrating the *LQG* controller for optimal closed-loop feedback based on measurements $y$ that are subjected to noise. The optimal LQR and *Kalman Filter* gain matrices $K_r$ and $K_f$ may be designed independently, based on two different algebraic Riccati equations. When combined, the resulting sensor-based feedback remains optimal. This controller is a dynamical system with an input $\mathbf{u}$, an output $\mathbf{y}$ and an internal state $\hat{\mathbf{x}}$. The main particularity of this controller is that the eigenvalues of the LQG system are given by the eigenvalues of the LQR and Kalman Filter gain matrices.
The entire framework of the LQG will be deepened, based on [19], in section 4.2.2 where it will be applied to control the attitude during the entire simulated maneuver, in particular the angular velocity vector and the quaternion vector with very good results.

## 1.3 Thesis Overview

In Chapter 2 the spacecraft mathematical model is introduced starting from the research work of Pirat [3]. In this chapter the dynamic equations for both attitude and position are stated to define the matrix representation of the state space equations needed for the state observation.

Chapter 3 focus on State Observer, starting from the introduction written in Chapter 1. The main applications of each observer and the mathematical model will be explained in detail, to individuate the parameters that should be set for the simulation.

In the second half of this thesis, the Control System Design (Chapter 4) used in the simulation will be addressed. In this chapter, lessons and frameworks learned from the control theory are presented by discussing their use and application in the case study. Once defined the algorithm, the setting parameters are individuate and tuned in Chapter 5

Finally, Chapter 5 will expose the entire simulation model. In the first section, the input data are defined, starting from the spacecraft characteristics from the simulation scenario. In the second half, the Attitude Control model, the Guidance and the Control parts are described in order to show the Control algorithm application results. In the last part, the Navigation algorithm is presented starting from the input data, needed by the Simulink model built for this research work, until the results given by the simulation. Numerical results are included, highlighting the advantages and drawbacks encountered in each observation application.

# Chapter 2

# Spacecraft Mathematical Model

## 2.1 Mission Description

The orbital simulator designed for this thesis work simulates a rendezvous and docking maneuver between two satellites: a Chaser and a Target. The satellite taken into account carries out two maneuvers before the final corridor: a Hohmann transfer and a Radial Boost. In the final phase, to align the docking ports, it starts the Cone of Approach Maneuver. In the following section these three phases are explained.

The local coordinate frame is used to describe motions with respect to the moving position and direction towards the center of the Earth of an orbiting body.

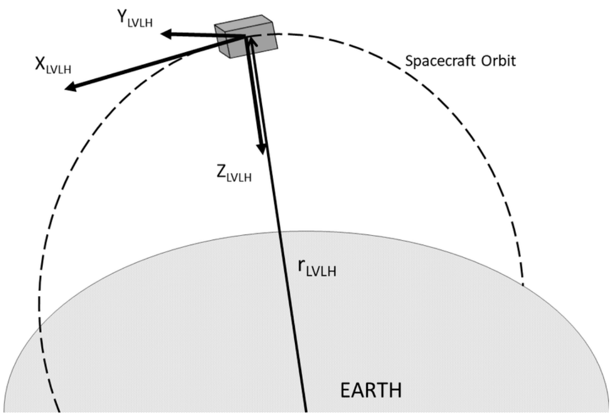


Figure 2.1. Local-vertical/local-horizontal frame [22].

In Figure 2.1 we have:

- The origin centered in the Centre of Mass of the spacecraft;
- $R_{bar}$ axis is along the vertical line towards the planetary center (Earth CoM);
- $V_{bar}$ is in the direction of the orbital motion parallel to the local horizontal;
- $H_{bar}$ is perpendicular to the orbital;

### 2.1.1 Hohmann Transfer

The Hohmann maneuver is carried out between two co-planar circular orbits and is defined as the most energy efficient two-pulse maneuver.

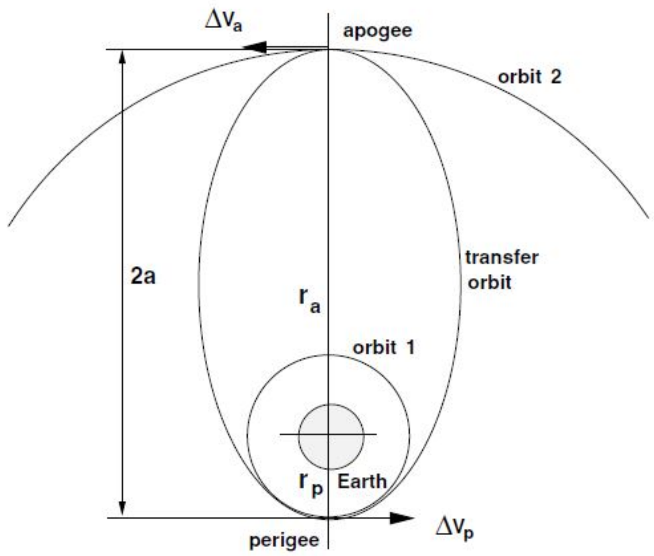


Figure 2.2. Orbit relations in a Hohmann transfer [22].

This consists of an elliptical transfer orbit tangent to the initial and final orbits. The starting point is at the perigee of the transfer orbit, the arrival point corresponds TO the apogee of the transfer orbit. The velocity has the same magnitude in the initial and final point of the maneuver and both are parallel to each other [22]. The $\Delta V$ is given parallel to the $V_{bar}$ axis and it can be calculated with the following equation:

$$\Delta V_{x1} = \Delta V_{x2} = \frac{\omega}{4} \Delta z \tag{2.1}$$

The input in this equation is represented by the Target angular velocity $\omega$ and by the difference between the Target and the Chaser altitude $\Delta z$. The first impulse

$\Delta V_{x1}$ is necessary both to leave the circular orbit and to join the elliptical Transfer Orbit (TO). The second one $\Delta V_{x2}$ is needed to keep the spacecraft on the Target orbit.

### 2.1.2 Radial Boost

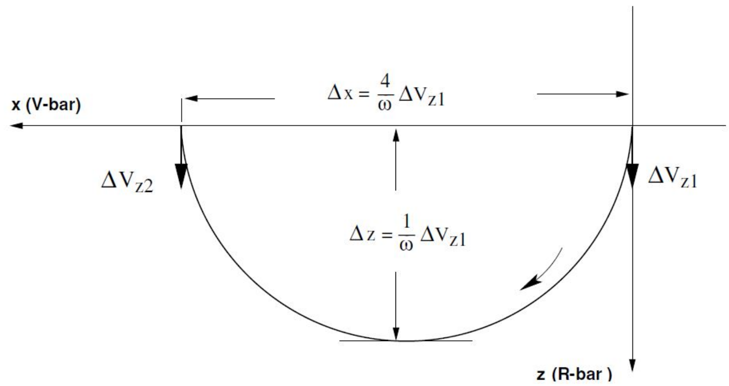


Figure 2.3. Transfer along V-bar by radial impulses [22].

Once the Chaser reaches the Target orbit, the new goal is to reduce the relative distance between the spacecrafts so that the final phase can start. In general, radial maneuvers are used for transfer along the Target orbit and for fly-around to an $R_{bar}$ approach.

In this maneuver the first boost is parallel to $R_{bar}$ and allows the Chaser to leave momentarily the orbit, going on lower (and so faster) one and returning on the Target orbit. Once here, a second and equal boost is given in the same directions of the first one to make the spacecraft remains on the initial orbit.

To calculate the $\Delta V_z$ it is necessary to know the Target angular velocity and the difference between the Target and the Chaser x-position $\Delta x$:

$$\Delta V_{z1} = \Delta V_{z2} = \frac{\omega}{4} \Delta x \tag{2.2}$$

The position of the final point before the final corridor depends on safety requirements, but it is usually located hundreds of meters far from the Target position. It is usually considered as transfer on $V_{bar}$ by radial impulses [22].

### 2.1.3 Final Approach

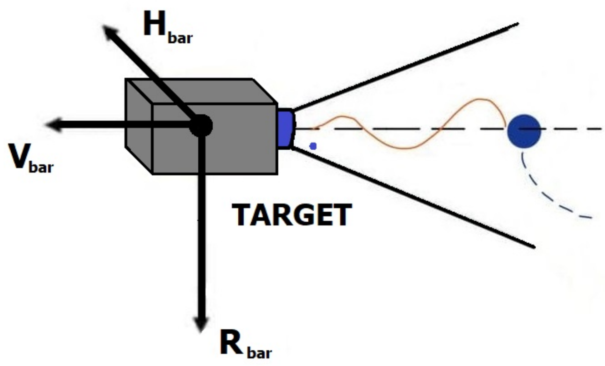


Figure 2.4. Final approach trajectory.

The final phase begin at the end of the radial maneuvers. The goal is to reach the position, velocity, attitude and angular rates condition to start the docking phase.

At this stage, the Chaser follows a quasi-straight line trajectory, but the main important thing is that the Chaser remains within a *cone of approach*, defined for safety reasons. The origin of this cone is the docking port of the Target vehicle, called mating point and the half cone angle is about 10 to 15 degrees. The Chaser must follow the docking axis, and this is only possible if the rendezvous sensors are able to measure the relative attitude between the docking port of Chaser and Target [22].

These proximity operations require extremely delicate maneuvering both in translational and rotational motion. In this phase the attitutde, angular rates, position, and velocity must be precisely controlled to obtain the required docking interface conditions.

### 2.1.4 Reference Mission

The trajectory profile for a CubeSat RVD mission provided in Figure 2.5 represents the approach strategies inspired by Pirat [3].

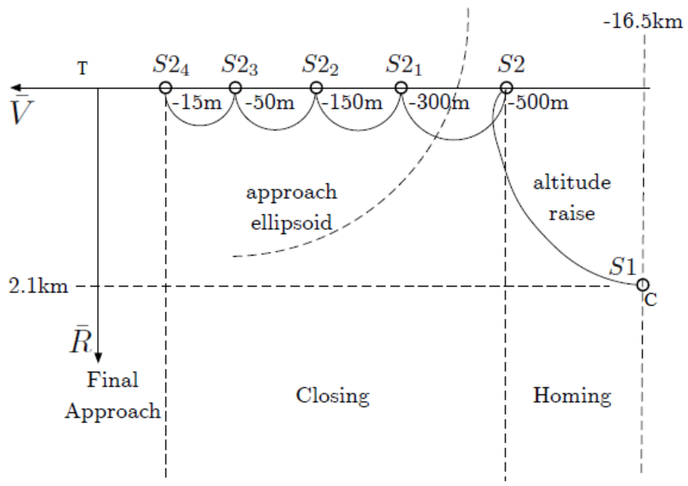


Figure 2.5. Trajectory profile for a CubeSat RVD mission [3].

During the first phase, the Homing is needed to allow the Chaser reaching the Target orbit. Thanks to an Hohmann transfer the Chaser can reach the point $S2$ and can start the radial maneuvers. In this strategy four different radial boosts along $V_{bar}$ are needed to reach the starting point of the final corridor.

Finally, from point $S2_4$ to the Target ($S3$) there are the final approach maneuvers. $S3$ location in the target orbital frame is defined by the target docking port location and orientation and thus it varies depending on the systems design. This point is always positioned 10 m away from the target docking port. In this phase it is really important to control the rotational and translational motion to avoid collision between satellites. The simulation of the entire maneuver will be further analyzed in Chapter 5.

## 2.2 Spacecraft Dynamics

The non-linear dynamic is developed and linearized for the simulation. This section represents a central part of the GNC to control the relative attitude and position between the Target and the Chaser. The coupling between the rotations and the translation will be taken into account.

Moreover, the position and the attitude dynamic will be fundamental for the Navigation, to have a comparison between the real value given by the dynamic and the estimated value provided by the state observation.

### 2.2.1 Rotational Motion

When determinating the attitude dynamic equations, it is important to know the geometry of the satellite. The first step is to calculate the body Inertial tensor **J** as:

$$J = \begin{bmatrix} J_{x0} & 0 & 0 \\ 0 & J_{y0} & 0 \\ 0 & 0 & J_{z0} \end{bmatrix} \tag{2.3}$$

The angular momentum of a rigid body, in inertial frame is defined as

$$\mathbf{H}_b = I_b \omega_b^{bI} \tag{2.4}$$

where $\omega_b^{bI}$ is the rotation of body reference frame in inertial reference frame expressed in body frame with the origin in the center of mass of the satellite. In torque equation the moments produced by the thrusters, the reaction wheels and the external disturbances must be taken into account. As concerning the evaluation of the angular rates, the classical *Euler equations* can be used.

$$\dot{\omega}_B = J^{-1}(M_B - \omega_B \times (J\omega_B + J_{RW}\omega_{RW})) \tag{2.5}$$

where

$$M_B = M_{thr} + \Delta M_{ex} + M_{RW} \tag{2.6}$$

Between the existent ways to represent or parameterize rotation, the quaternion representation is chosen. This representation has several advantages over Euler's angle. First of all the less computational effort given not by the derivative of the Euler's angle, but by the dependence from the angular velocities. The second reason is that there is no geometric singularity. These motivations make quaternion more effective and stable numerically [39].

The evolution of the quaternions is described by the set of linear differential equation is represented in the matrix form as:

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \tag{2.7}$$

The equivalent matrix form is given by

$$\dot{q}_0 = -\frac{1}{2} q_v^T \omega_B$$

$$\dot{q}_v = \frac{1}{2}(q_0 \mathbb{1} + Q^x)\omega_B \tag{2.8}$$

where $q_v = [q_1, q_2, q_3]$, and $Q^x$ is the skew-symmetric matrix defined as:

$$Q^x = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{2.9}$$

When writing the state space system for the attitude dynamic, the state vector **x** and the control torque **u** need to be defined.

$$x = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \qquad u = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \tag{2.10}$$

Then the state matrix **A** and the input matrix **B**

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \omega_0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\omega_0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -\frac{3\omega_0}{J_{x0}} & 0 & 0 & 0 & -\frac{(J_{z0}-J_{z0})\omega_0}{J_{x0}} \\ \frac{3\omega_0}{J_{y0}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(J_{y0}-J_{x0})\omega_0}{J_{z0}} & 0 & 0 \end{bmatrix} \tag{2.11}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1/J_{x0} & 0 & 0 \\ 0 & 1/J_{y0} & 0 \\ 0 & 0 & 1/J_{z0} \end{bmatrix} \tag{2.12}$$

From [21] the linearized nadir pointing spacecraft model with gravity gradient disturbance torque and magnetic torque are taken into account and used for the simulation:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \\ f_1 & 0 & 0 & 0 & 0 & f_2 \\ 0 & f_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & f_4 & f_5 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ M_x/J_{x0} \\ M_y/J_{y0} \\ M_z/J_{z0} \end{bmatrix} \tag{2.13}$$

where:

$$f_1 = 8(J_{z0} - J_{y0})\omega_0^2/J_{x0} \tag{2.14}$$

$$f_2 = (-J_{x0} + J_{y0} - J_{z0})\omega_0/J_{x0} \tag{2.15}$$

$$f_3 = 6(J_{z0} - J_{x0})\omega_0^2/J_{y0} \tag{2.16}$$

$$f_4 = 2(J_{x0} - J_{y0})\omega_0^2/J_{z0} \tag{2.17}$$

$$f_5 = (J_{x0} - J_{y0} + J_{z0})\omega_0/J_{z0} \tag{2.18}$$

### 2.2.2 Translational Motion

The aim is to represent the Chaser motion in the rotating target local orbital frame, which find its origin in the center of mass of the Target in LVLH frame. The Hill's equations describe a simplified model of the orbital relative motion. These equations consider both the Target and the Chaser placed in a circular orbit.

This model gives a first-order approximation of the Chaser motion in a Target-centered coordinate system. These equations can be considered when the distance between the Chaser and the Target is lower than the orbital radius. The equations are:

$$\begin{cases} \ddot{x} = \frac{1}{m_c}F_x + 2\omega\dot{z} \\ \ddot{y} = \frac{1}{m_c}F_y - \omega^2 y \\ \ddot{z} = \frac{1}{m_c}F_z - 2\omega\dot{x} + 3\omega^2 z \end{cases} \tag{2.19}$$

The forces are given by the thrusters, which control is given by the control system. The variable mass is calculated through the Tsiolkovsky equation:

$$\dot{m} = \frac{F}{gI_{sp}} \tag{2.20}$$

and $\omega$ is the constant orbital angular velocity.

$$\omega = \sqrt{\frac{\mu}{r_t^3}} \tag{2.21}$$

As highlighted in (2.19), there is a coupled dynamicS between x and z axis, while the y axis is decoupled from the other variables. For this reason, once linearized, the dynamic can be easily studied separately.

As regard the x and z dynamic, defining the state vector $x = [x, z, \dot{x}, \dot{z}]^T$ and the control force $u = [F_x, F_z]$, from [22], the state space equation is:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{2.22}$$

The A and B matrices are:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2\omega \\ 0 & 3\omega & -2\omega^2 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m_c & 0 \\ 0 & 1/m_c \end{bmatrix} \tag{2.23}$$

Whereas the position along the y axis, the state vector is $x = [y, \dot{y}]$, the control force $u = F_y$, and the matrices A and B are:

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1/m_c \end{bmatrix} \tag{2.24}$$

These are the Hill's equations written in the state space form used as input for the state observers.

## 2.3 External Disturbances

During the simulation, some external disturbances are taken into account, considering the altitude, the velocity and the dimensions of the Chaser. The Chaser orbit has an altitude of 397.9 km, while the Target one of 400 km. Thus, these orbits can be defined as Low Earth Orbit (LEO) and from [40], the disturbances that must be considered in these orbits are:

- **Aerodynamic Drag**:

  This drag derives from some residual atmosphere existent in LEOs. The conventional fluid mechanics can't be applied because of the poor density of the atmosphere. In fact, the drag phenomenon must be treated at the molecular level.

  Actually, there are two different atmospheric models used to compute the atmospheric density, and these depend on the altitude. The drag coefficient is also dependent from the impact of the molecules on the surface of the spacecraft and their reflection.

  To calculate the aerodynamic drag we use:

  $$\bar{T} = \bar{r} \times \bar{F_A} \tag{2.25}$$

  where

  $$F_A = \frac{1}{2}\rho V^2 S C_D \tag{2.26}$$

  and $\bar{r}$ is the vector from body center mass to aerodynamic Center of Pressure, so it varies with the attitude, $\rho$ is the air density, $V$ is the velocity of the satellite, $S$ is the front Area and $C_D$ is the drag coefficient.

If $C_D = 2.2$ is considered, the obtained value of the air drag has a small error, whose value is taken into account. For low orbit satellites, the air density is high enough to produce a perturbing force, modeled as a constant force into simulation of the entire maneuver.

- **Gravity Gradient**:

  In space, the gravitational field is not considered uniform, and the variation in the magnitude and direction of the gravitational force over a spacecraft, leads to a gravitational torque about the body mass center.

  Satellites move around the gravity field of a central body. If we consider the central body as a perfectly spherical the satellites will adhere to Newtonian laws.

  The generated torque is perpendicular to local vertical and can be calculated as:

$$\bar{T} = 3n^2 \cdot \hat{r} \times [\mathbf{I}\hat{r}] \tag{2.27}$$

  where $n$ is the orbital rate and $\mathbf{I}$ is the inertia tensor of the spacecraft. An important aspect of the gravity gradient torque is that it can be used as a method of attitude control using this effects.

- **Magnetic Torque**:

  Another important disturbance affecting the attitude control is the Earth magnetic field influence. Last interacts with the electrical currents inside the satellite. Thus, the magnetic torque can be calculated as:

$$\bar{T} = \bar{M} \times \bar{B} \tag{2.28}$$

  where M is the spacecraft residual dipole measured in $A/m^2$. This is due to current loops and residual magnetization. It's value is at least 100 $A/m^2$ or more for a CubeSat. $B$ is the Earth magnetic field vector measured in Tesla or Gauss. This value decrease as $1/r^3$ with its direction along magnetic field line.

# Chapter 3

# State Observer

Starting from the introduction provided in the first chapter, the following section introduces the mathematical model of each observer so as to define the equation necessary for the simulation.

## 3.1 Kalman Filter

The Kalman Filter (KF) is a type of state observer, but it is designed for stochastic systems. Analyzing the Kalman Filter equation:

$$\hat{x} = A\hat{x}_{k-1} + Bu_k + K_k(y_k - C(A\hat{x}_{k-1} + Bu_k)) \tag{3.1}$$

it is possible to notice how the Kalman filter equation relates to the probability density function. The first part predicts the current state by using state estimate from the previous time step and the current input and can be called *a priori estimate*. The second part uses measurement and it is called *a posteriori estimate*.

The Kalman Filter computations are based on five equations that will be further explained in the following section. There are two *prediction equations*, the *Kalman Gain Equation* and finally, two *update equations* [23].

The first two prediction equations are:

- **State Extrapolation Equations**

  The Kalman Filter bases its predictions or estimations on the known present estimation of the actual state. Thanks to this equation, the next system state can be predicted by knowing the current state. It extrapolates the state vector from the current step (step $n$) to the future one (step $n+1$).

  In a matrix notation the state exploration equation can be expressed/is expressed as:

$$\hat{x}_{n+1,n} = F\hat{x}_{n,n} + G\hat{u}_{n,n} + w_n \tag{3.2}$$

where:
$\hat{x}_{n+1,n}$ is the predicted vector at time step $n+1$
$F$ is the *State transition matrix*
$\hat{x}_{n,n}$ is the estimated vector at time step $n$
$G$ is the *Input transition matrix*
$\hat{u}_{n,n}$ is the control variable that is the measurable input for the system
$w_n$ is the process noise. This is an unmeasurable value that affects the process.

- **Covariance Extrapolation Equation**

This equation represents the uncertainty in the prediction and the general form of the Covariance Extrapolation Equation is given by:

$$P_{n+1,n} = FP_{n,n}F^T + Q \tag{3.3}$$

where:
$P_{n+1,n}$ is the predicted covariance matrix for the next state
$P_{n,n}$ is the estimated covariance matrix of the current state
$Q$ is the process noise matrix

The third equation is:

- **Kalman Gain Equation**

This gain is needed for the computation of both the update equations. It defines the weight of the past estimation and the weight of the measurement in estimating the current state. This number, which has a value between 0 and 1, is defined by :

$$K_n = P_{n,n-1}H^T(HP_{n,n-1}H^T + R_n)^{-1} \tag{3.4}$$

where:
$K_n$ is the Kalman Gain;
$P_{n,n-1}$ is a prior estimate uncertainty matrix of the current state predicted at the previous state
$H$ is the *Observation matrix*
$R_n$ is the measurement noise covariance matrix

The final equations are:

- **State Update Equation**

  Thanks to the known past estimation and the current measurement, the estimation of the current state is calculate. The matrix form of this equation is:
  $$\hat{x}_{n,n} = (1 - HK_n)\hat{x}_{n,n-1} + K_n z_n \tag{3.5}$$
  Defining

  | | |
  |---|---|
  | $\hat{x}_{n,n}$ | is the estimated state at time step $n$ |
  | $H$ | is the *Observation matrix* |
  | $\hat{x}_{n,n-1}$ | is the predicted state vector at time step $n-1$ |
  | $z_n$ | is the measurement |

  It can be said that the Kalman Gain $K_n$ is the weight given to the measurement and $(1 - K_n)$ is the value given to the estimation. So, when we have both small measurement uncertainty and large estimate uncertainty, we must chose a Kalman gain close to one. Vice versa, we chose a Kalman gain close to 0. If the estimate uncertainty is equal to the measurement uncertainty $K_n$ is close to 0.5.

- **Covariance Update Equation**

  This represenst the uncertainty in the prediction.

  $$P_{n,n} = (I - K_n H)P_{n,n-1}(I - K_n H)^T + K_n R_n K_n^T \tag{3.6}$$

  Where:
  $P_{n,n}$ is the current state covariance uncertainty
  $P_{n,n-1}$ is a prior estimate uncertainty matrix of the current state predicted at the previous state
  $R_n$ is the measurement noise covariance matrix

## 3.2 Extended Kalman Filter

Generally, most realistic problems involve nonlinear function. The main question is how to take into account this non-linearity. An effective solution can be found getting rid of these two expressions:

$$\dot{x}_t = Ax_t + Bu_t \qquad\qquad y_t = Cx_t + Du_t \tag{3.7}$$

and turn them into new functions:

$$\dot{x}_t = g(u_t, x_{t-1}) \qquad\qquad y_t = h(x_t) \tag{3.8}$$

By using non-linear functions to perform the observation, it is necessary to leave the Gaussian assumption because if we put in a Gaussian through a non-linear function, the output function will not be a Gaussian anymore. Therefore, noticed that the Gaussian assumption is not respected, the Kalman Filter is not applicable anymore.

With the aim of finding a solution to this matter, a non-linear function is assumed as a linear one and is locally linearized. This method is called *local linearization* and basically takes a point in which the first derivative of this function is computed, fitting a line through this non-linear function by computing Taylor expansion. Hence looking at the Prediction and Correction models:

- Prediction:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) \tag{3.9}$$

- Correction

$$h(x_t) \approx h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t) \tag{3.10}$$

The Jiacobian matrices can be defined as:

$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}} \qquad\qquad H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \tag{3.11}$$

In the first equation there is the non-linear function $g$, which is influenced by the parameters $u$ and $x_{t-1}$. Specifically, $x_{t-1}$ is defined by the evaluation of the function at the linearization point, plus the Jacobian of the first derivative of this function with respect to the state x and the therm $\mu_{t-1}$, determining in return how far the variable $x_{t-1}$ is far from the linearization point.

Now, as to the correction step there are both: the dependence from $x_t$ and from the linearization point $\bar{\mu}_t$. The latter is found to be in line with the predicted scenario, seen that the best estimate possible concerns the state before executing the correction. Subsequently, there is the Jacobian matrix $H_t$, which is the first derivative of the function $h$, with respect to $x_t$ multiplied by the therm indicating how far it is from the linearization point.

The Jacobian is a non-square matrix $m \times n$ filled with partial derivative representing the orientation of the tangent plane to the vector-valued function at a given point.

As a result of the Taylor expansion, $G_t$ and $H_t$ turn into linear functions. The next questions that need to be answered regard the breadth of the error done computing the linearization, and the reasons determining it. A possible solution to the latter can be found in relation to the two quantities from whom it actually depends.The first quantity refers to the approximation error and shows not only how far is the linearized function from the initial function, but also the success of the local linearization. The second one is the uncertainty: the smaller is the value of the uncertainty, the smaller the error of the linearization.

As regard as the algorithm, it is basically the same of the classical Kalman Filter, excluding the replacing of the linear model with the non-linear function for the prediction of the mean, seen that it predicts just one single point. The same holds for the observation, through the use of a function capable of comparing what is actually observed and what is predicted. Another difference with respect to the KF is given by the substitution of the matrix $A$ with the matrix $G$, as well as the matrix $C$ with the matrix $H$, which are respectively the Jacobian of the motion and the observation.

Finally, it can be said that the Kalman Filter works well until extremely large uncertainties are not involved, because that could subsequently cause large errors into the linearization. The second issue that could eventually emerge would be the failure of the Kalman filter, caused by the linearized function's wrong approximation of the non-linear models.

## 3.3 $1^{st}$ order Sliding Mode Observer

An observer is essentially a mathematical replica of the system, driven by the system input and the estimation error. A sliding mode observer (SMO) feedback the output estimation error by adopting a non-linear switching term. Once the magnitude of the disturbance is known, this observer forces both the output errors to converge to zero in a certain amount of time, and the observed states converge to the system state.

Starting from the state space representation of a linear system:

$$\dot{x}(t) = Ax(t) + Bu(t) + G_n\nu \tag{3.12}$$

$$y(t) = Cx(t) + Du(t) \tag{3.13}$$

the goal is to estimate $x(t)$ basing only on the quantities $y(t)$ and $u(t)$ [1]. When writing the Sliding Mode Observer model, it is important to define the state estimation and the output estimation error, respectively defined as

$$\epsilon_x = \hat{x} - x \tag{3.14}$$

$$\epsilon_y = \hat{y} - y \tag{3.15}$$

In many cases the objective is to estimate the velocity $x_2$. In order to estimate $x_2$ the SMO algorithm is the following:

$$\dot{\hat{x}}_1 = \nu \tag{3.16}$$

where $\nu$ is the *observer injection term* that is necessary to be designed, so that the estimates $\hat{x}_1$ and $\hat{x}_2$ will converge to $x_1$ and $x_2$.

Therefore, $\nu$ is defined as

$$\nu = -\rho sign(z_1) \tag{3.17}$$

where the estimation error, that is also the auxiliary sliding variable, is

$$z_1 = \hat{x}_1 - x_1 \tag{3.18}$$

and

$$\dot{z}_1 = -x_2 + \nu_{eq} = 0 \tag{3.19}$$

The sliding mode observer is computed using the concept of the *equivalent control.* This concept refers to the control function that needs to be applied to the system, to ensure the system trajectory stays on the surface. The equivalent control action describes the “average” effect of the high-frequency switching control on system [1]. From 3.3 the estimated second state is defined as:

$$x_2 \approx \hat{x}_2 = \hat{\nu}_{eq} \tag{3.20}$$

To estimate $x_2$, it is important to define the *equivalent injection term.* The latter can be calculated thanks to a Low Pass Filter:

$$\tau\dot{\hat{\nu}}_{eq} = -\hat{\nu}_{eq} - \rho sign(z_1) \tag{3.21}$$

thus, as it can be read in 3.3, the estimated second state is totally dependent from $\nu_{eq}$ . One of the most critical disadvantages concerns the accuracy of the observation, that decreases once the step time of the simulation rises. More the step time is bigger, more the estimation of the state variable is worst.

This derives both from an imperfection in the sign function producing a finite amplitude and finite frequency “zigzag” motion, and from the discrete-time nature of the computer simulation. This phenomenon is called *chattering* and there are several methods capable of solving it.

The one selected for this research is the *quasi-sliding mode*, in which the sign function is substituted by the sigmoid function, defined as follow:

$$sign(\sigma) = \frac{\sigma}{|\sigma| + \varepsilon} \tag{3.22}$$

The introduction of this function removes the zigzag pattern with a continuous one.

Hence, the parameters that affect the good performance of the observer that must be tuned are $\rho$ and $\tau$. $\rho$ is bounded as

$$\rho = |x_2| + \beta \qquad\qquad \beta > 1 \tag{3.23}$$

If we consider the quasi sliding mode, the third parameter that must be tuned is $\varepsilon$. From [1] it can be said that the SMO also could be treated as differentiator, since the variable it estimates is the derivative of the measured variable.

## 3.4 Super-Twisting Sliding Mode Observer

Frequently, linear observers do not achieve adequate performances for such systems [24]. The sliding mode observers have main advantages like their robustness, with respect to uncertainties, and the finite-time convergence. The observers based on the second order sliding mode, require the proof of the *separation principle* theorem. This is due to the asymptotic convergence of the estimated values to the real values.

The separation principle is a control theory assumption and states that, under some presumption, the problem given by the design of a optimal feedback controller for a stochastic system, can be solved thanks to the design of an optimal observer of the state variable of the system. This combination feeds into an optimal deterministic controller for the system [25]. Thus, the design can be facilitated by the division of the problem into two separate parts.

Thanks to [1], [24] and [26] this research felt on the Super Twisting algorithm because its implementation doesn't need the separation principle to be proved [24]. The observer proposed in this thesis is the second order Sliding Mode Super-Twisting algorithm. This observer is used for the estimation of position, velocity and attitude variables. The *Super Twisting Sliding Mode Observer* (STO) has the following form:

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + z_1 \\ \dot{\hat{x}}_2 &= f(t, x_1, \hat{x}_2, u) + z_2 \end{aligned} \tag{3.24}$$

In 3.24, $\hat{x}_1$ and $\hat{x}_2$ can be defined as the state estimation terms which can be calculated thanks to the correction variables $z_1$ and $z_2$, which are the output injection terms defined by

$$\begin{aligned} z_1 &= \lambda|x_1 - \hat{x}_1|^{1/2} sign(x_1 - \hat{x}_1) \\ z_2 &= \alpha sign(x_1 - \hat{x}_1) \end{aligned} \tag{3.25}$$

In chapter 1 it has been defined:

$$\begin{aligned} \varepsilon_1 &= x_1 - \hat{x}_1 \\ \varepsilon_2 &= x_2 - \hat{x}_2 \end{aligned} \tag{3.26}$$

By substituting 3.26 in 3.25, and 3.25 in 3.24, it can be obtained the error equations:

$$\begin{aligned} \dot{\hat{x}}_1 &= \hat{x}_2 + \lambda|\varepsilon_1|^{1/2} sign(\varepsilon_1) \\ \dot{\hat{x}}_2 &= F(t, x_1, \hat{x}_2, u) + \alpha sign(\varepsilon_1) \end{aligned} \tag{3.27}$$

where $F(t, x_1, \hat{x}_2, u)$ is defined as:

$$F(t, x_1, \hat{x}_2, u) = f(t, x_1, \hat{x}_2, U(t, x_1, x_2)) + \xi(t, x_1, x_2, U(t, x_1, x_2)) \tag{3.28}$$

Therefore, it must be supposed a bounded system, which existence is ensured by the constant $C$ through the inequality $|F(t, x_1, \hat{x}_2, u)| < C$.

The estimation constant $f^+$ can be found as the double maximal possible acceleration of the system [24]. Moreover, also $\alpha$ and $\lambda$ must satisfy the following inequalities:

$$\begin{aligned} &\alpha > C \\ &\lambda > \sqrt{\frac{2}{\alpha - C}} \frac{(\alpha + C)(1+p)}{1-p} \end{aligned} \tag{3.29}$$

where $p$ is a chosen constant, $0 < p < 1$.

$\alpha$ and $\lambda$ are defined as:

$$\alpha = a_1 C \qquad \lambda = a_2 \sqrt{C} \tag{3.30}$$

For [24] a valid choice is $a_1 = 1.1$ and $a_2 = 1{,}5$ so

$$\begin{cases} \alpha = 1.1C \\ \lambda = 1.5\sqrt{C} \end{cases} \tag{3.31}$$

In conclusion, the only parameter that has to be tuned to adjust the observation is $C$. Moreover, the mathematical model of the dynamic of the system must be written in order to define $\dot{\hat{x}}_2$ and then calculate $\dot{\hat{x}}_1$.

# Chapter 4

# Control System Design

Control theory deals with the control of dynamic systems in processes and engineered machines. The objective is to develop a model or algorithm that regulates the application of system inputs to bring the system to the desired state, minimizing any delays, delay, overshoot, or steady-state error and ensuring a level of control stability; often with the aim of achieving a degree of optimization. To do this, is needed a controller with the necessary corrective behavior. This controller monitors the controlled process variable and compares it to the desired value. The difference between the actual value and the desired value of the process variable, called error signal, is applied as feedback to generate a control action to bring the controlled variable to the same value as the desired variable.

Other aspects, which are also studied, are the controllability and the observability. This is the basis for the advanced type of automation that revolutionized manufacturing, aircraft, communications and other industries. This is feedback control, which involves taking measurements using a sensor and making calculated adjustments to keep the measured variable within a set range by means of a final control element.

## 4.1 Translational motion control

For the translational motion the following controller are adopted to control the position along $V_{bar}$, $R_{bar}$ and $H_{bar}$.

### 4.1.1 PID controller

A PID controller is an instrument used in several applications to regulate process variables. The *Proportional Integral Derivative* controllers use a control loop feedback mechanism to control process variables, which are the most accurate and stable controller. PID control is a well-established way of driving a system towards a target position or level. It is the most common, as a means of controlling variables and finds application, in myriad scientific processes as well as automation. PID control uses closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible [31]. The overall control function is:

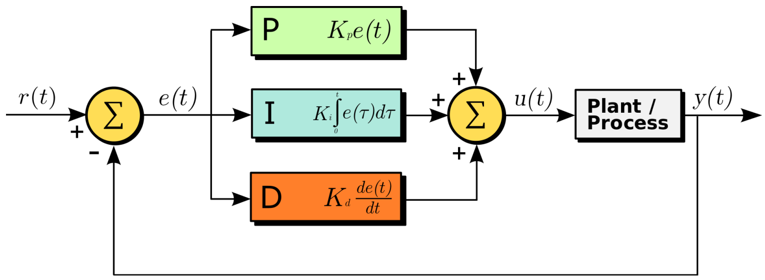


Figure 4.1. PID in a feedback loop [33].

$$u(t) = K_p e(t) + K_i \int_0^t e(t')\, dt' + K_d \frac{de(t)}{dt} \tag{4.1}$$

The **proportional term** depends only from the difference between the set point and the process variable. This difference is defined as *error term* $e(t)$. The proportional gain $K_p$ determines the ratio of the output response to the error signal. In general, increasing the proportional gain, there will be an increase of the speed of the control system response. However, if the proportional gain is too large, the process variable will begin to oscillate [32]. If the gain is further increased, there will be larger oscillations and the system will become unstable and out of control.

The **integral term** sums the error term over time. Consequently, also a small error will increase the integral term slowly during the evolution of the control. The integral response will increase over time unless the error is zezo, hence the objective is to reach the target value and drive the *Steady-State error* to zero. The Steady-State error is the difference between the final value of the controlled variable and the target [31].

The **derivative term** causes the decrease of the output if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable [32]. By increasing the derivative gain $K_d$, the system will react more strongly to changes in the error term and will increase the response speed of the system. This term is highly sensitive to noise, in the process variable signal [31], indeed, if the feedback signal is noisy, the derivative response could make the control unstable.

### 4.1.2 Sliding Mode Control

The *Sliding Mode Control (SMC)* is a control methodology used for non-linear systems, it has very solid theoretical foundations (developed by Russians engineer) and has an important characteristic: if the system is not well known or is known within certain uncertainties, the control works the same. This is a "robust" control system to uncertainties, indeed it is one of the few robust non-linear controls.

If $x$ is the usual vector of states and to describe the system, the representation in state variables is used, where the derivative of the vector of states turns out to be, in general, a function of the input force $u(t)$ and the same state $x(t)$, so that

$$\begin{cases} \hat{x}_1 = x_2 \\ \hat{x}_2 = u + f(x_1, x_2, t) \end{cases} \tag{4.2}$$

First, a new variable in the state space form of the system must be introduced:

$$\sigma = \sigma(x_1, x_2) = x_2 + cx_1, \qquad c > 0 \tag{4.3}$$

where $\sigma$ is the *Sliding Manifold.* This last is a vector function $\sigma(x) = [\sigma_1(x), \sigma_2(x)]_R^n \to R^m$. In the simplest case $n = 2$ and $m = 1$, so there are two states $x(t) = [x_1(t), x_2(t)]$ and the motion will be in a Cartesian plane. Thus, the function $\sigma(x(t)) = x_1 + cx_2 = 0$ represents the equation of a line passing through the origin in the plane $(x_1 - x_2)$. In the simplest case, it can be defined as a plane, in the multidimensional systems this is better defined as a surface [34].

Figure 4.2 represents the manifold in the 2D plane as a line. In the tridimensional plane, if $\sigma_1 = 0$ and $\sigma_2 = 0$, an intersection of two planes is obtained and then $\sigma$ is a straight line. The aim is to start from the initial position, get to the manifold and then slide on the manifold itself. The control law creates a way to bring the system on the sliding surface and allows it to slide on the same. In this way, the control law keeps the system on the surface, correcting the latter every time it moves away from the sliding surface, which in turn represents the ideal model

In order to achieve asymptotic convergence of the state variables it is necessary to drive the variable $\sigma$ in Eq. 4.3 to zero in finite time by means of the control u. This task can be achieved by applying Lyapunov function techniques to the $\sigma$-dynamic which are derived using Eqs. 4.2 and 4.3 [27]:

$$\dot{\sigma} = cx_2 + f(x_1, x_2, t) + u \tag{4.4}$$

The $\sigma$-dynamic is defined, afterwards the Lyapunov function is introduced in the form [28]

$$V = \frac{1}{2}\sigma^2 \tag{4.5}$$

The control law $u$ driving $\sigma$ to zero in finite time is:

$$u = -cx_2 - \rho sign(\sigma) \tag{4.6}$$

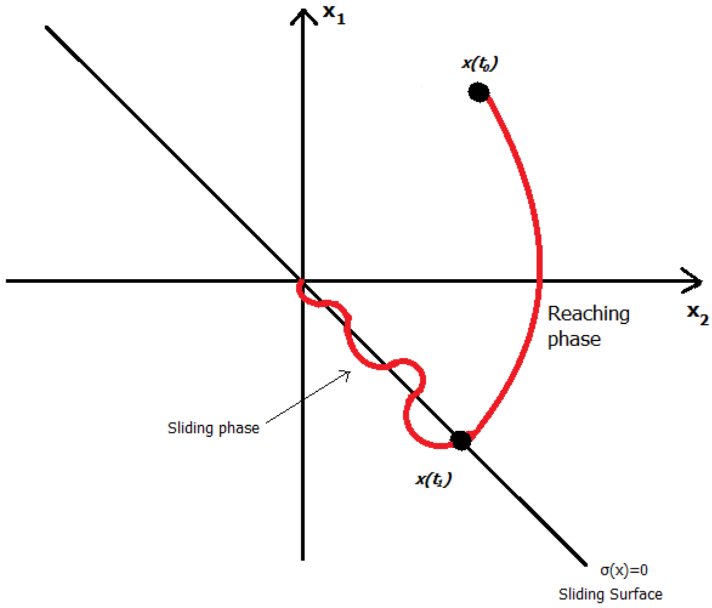


Figure 4.2. 2D Sliding surfaces [27].

where $\rho$ is a control gain. The mathematical model is well explained in [27] and [28] but it does not fall within the scope of this thesis.

The main disadvantage is the *chattering* phenomenon explained in 3.3. The proposed and used method to occur this phenomenon is the *Quasi-Sliding Mode*, which substitute the sign function with the *Sigmoid function* 3.22.

## 4.2 Rotational motion control

As regard the rotational motion the Quaternion Feedback Control is developed for a comparison with the LQG controller. The mathematical model for each observer is shown in the following two section.

### 4.2.1 Quaternion Feedback Control

Satellites are often reoriented performing successive rotations around the control axis, with the aim of achieving the desired attitude. However, this strategy is not optimal in terms of fuel or energy consumption. The *Quaternion Feedback Control (QFC)* gives a nearly-optimal orientation with a very simple control loop, easy to implement.

The QFC results are used as a comparison with the results provided by the Linear Quadratic Gaussian controller, and with the classical LQR combined with the Super Twisting Observer.

With the aim of achieving a good control, it is necessary to use a quaternion vector $q_e$. This latter can be defined as following:

$$q_e = q_{true} - q_{des} \tag{4.7}$$

therefore, this is not a simple difference, but is computed through [30]

$$q_e = q_{des}^{-1} \otimes q_{true} \tag{4.8}$$

where

$$q_d = q_{des}^{-1} = \frac{q_{des}^*}{||q_{des}||_2} = \frac{[q_0 - q_1 - q_2 - q_3]^T}{(q_0^2 + q_1^2 + q_2^2 + q_3^2)} \tag{4.9}$$

the quaternion product result is:

$$q_e = \begin{bmatrix} q_{d0} & -q_{d1} & -q_{d2} & -q_{d3} \\ q_{d1} & q_{d0} & -q_{d3} & q_{d2} \\ q_{d2} & q_{d3} & q_{d0} & -q_{d1} \\ q_{d3} & 0 - q_{d2} & q_{d1} & q_{d0} \end{bmatrix} \cdot \begin{bmatrix} q_{true0} \\ q_{true1} \\ q_{true2} \\ q_{true3} \end{bmatrix} \tag{4.10}$$

Once the the desired attitude is bounded, it must be assumed that the inertial reference frame coincides with the desired attitude, so that $q_d = [1{,}0{,}0{,}0]^T$ and $q_e = q_v$, with only the vectorial part $q_v = [q_1, q_2, q_3]^T$. The unitary quaternion represents an attitude perfectly alligned with the desired reference frame.

The control torque is defined as

$$M_B = -sign(q_{0,d}) K_p q_v - K_d \omega_b \tag{4.11}$$

$K_p$ and $K_d$ are gain matrices used as stiffness and damping coefficients.

For this simulation the gain $K_d$ is defined as:

$$K_d = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \tag{4.12}$$

As regard as $K_p$, two definition are proposed and tested:

$$K_p = ksign(q_{true0})\mathbb{1} \tag{4.13}$$

$$K_p = [\alpha \mathbf{I} + \beta \mathbb{1}]^{-1} \tag{4.14}$$

Where $k$ and $d_i$ are positive scalar constants, $\mathbb{1}$ is the $3 \times 3$ identity matrix, $\mathbf{I}$ is the $3 \times 3$ inertia matrix, $sign(q_{true0})$ is the sign function, and $\alpha$ and $\beta$ are non negative scalars.

### 4.2.2 Linear Quadratic Gaussian Controller

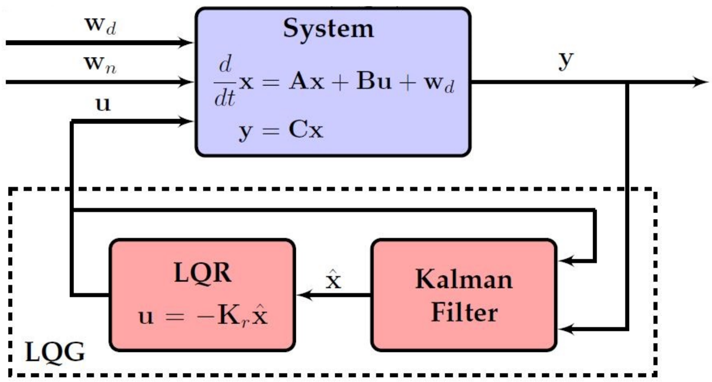


Figure 4.3. Linear Quadratic Gaussian [19].

The *Linear Quadratic Gaussian* (LQG) is the result of the combination between the Linear Quadratic Regulator (LQR) full-state feedback and Kalman Filter full-state estimator. The LQG controller is a dynamical system with the input $y$, the output $u$ and the state $\hat{x}$ that follow the equation:

$$\dot{\hat{x}} = (A - K_f C - BK_r)\hat{x} + K_f y \tag{4.15}$$

$$u = -K_r\hat{x} \tag{4.16}$$

The LQG follow the optimal cost function of the LQR:

$$J(t) = \int_0^t [x(t) \cdot Qx(t) + u(t) \cdot Ru(t)]dt \tag{4.17}$$

The controller $u$ in 4.16 depends of the state estimation, so the cost function must take into account the disturbance and noise in the process. Therefore, the state space representation equation must include $w_d$ and $w_n$which correspond respectively to the disturbance and to the noise in the process:

$$\hat{x} = Ax + Bu + w_d \tag{4.18}$$

$$y = C_x + w_n \tag{4.19}$$

Substituting 4.16 in 4.18, including the identity $\hat{x} = x - (x - \hat{x})$, the result is

$$\dot{x} = Ax - BK_r x + BK_r(x - \hat{x}) + w_d \tag{4.20}$$

Now the the dynamic of the estimation error $\varepsilon = x - \hat{x}$ must be introduced to combine those equations:

$$\dot{\varepsilon} = (A - K_f C)\varepsilon + w_d - K_f w_n \tag{4.21}$$

The gain matrices $\mathbf{K_r}$ and $\mathbf{K_f}$ may be designed independently, based on two different algebraic Riccati equations. When combined, the resulting sensor-based feedback remains optimal [19].

Finally, combining 4.20 and 4.21, the closed-loop system can be defined as:

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A - BK_r & BK_r \\ 0 & A - K_f C \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix} + \begin{bmatrix} I0 \\ I - K_f \end{bmatrix} \begin{bmatrix} w_d \\ w_n \end{bmatrix} \tag{4.22}$$

Thereby, the dynamic of $x$ and $\varepsilon$ can be derived. As a consequence, the eigenvalues of this coupled system are given by the eigenvalues of $A - BK_r$ and $A - K_f C$ because the matrix is diagonal. Thus, the main advantage happens when those systems are combined, the eigenvalues of the full-state $x$ are stabilized by the LQR and the eigenvalues of the estimation dynamic are stabilized by the Kalman Filter [34]. In control theory, this is called *Separation Principle* which essentially means that the controller and estimator can be designed separately, and once combined, the coupled system retains the properties of each system.

This controller will be applied to control the attitude variables such quaternions and angular velocities in chapter 5. Moreover, the combination between the LQR and the other state observer studied in Chapter 3 will be tested to understand which estimator can provide best results.

# Chapter 5

# Mission Scenario - Simulation and Results

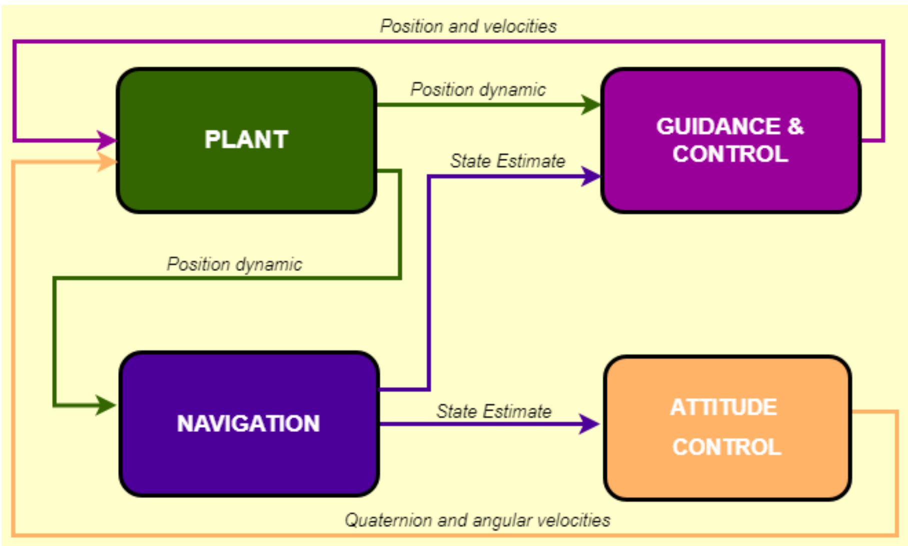


Figure 5.1. Orbital simulator block diagram.

In figure 5.1 it is possible to have a look at the representation of the orbital simulator block diagram developed during the thesis work. The simulator is developed on Simulink, while the input data are taken from a Matlab code. Basically, it is a classical Guidance, Navigation and Control scheme in which the attitude and position control are separated, although they work simultaneously. In the following chapter, the block diagram will be deepened and the results of the simulations are shown.

Starting from the plant, it contains the position dynamic equation shown in section 2.2.2. For simplicity, into the simulation, the attitude dynamic equations are calculated in the attitude block, in which the control is also executed. In the guidance block, all the maneuver are accomplished to reach each way-point shown in figure 2.5. In the Navigation Block, all the state observer studied in chapter 3 are implemented in order to estimate each state variables involved in the simulation.

## 5.1 Input Data

In this section, all the input data given to the Simulink code are shown, in order to simulate the entire maneuver.

### 5.1.1 Orbital Characteristics

The maneuver is accomplished in Low Earth Orbit. The Chaser orbit is lower than the Target one, in particular:

$$
\begin{aligned}
z_{chaser} &= 397.900\, m \\
z_{target} &= 400.000\, m
\end{aligned}
\tag{5.1}
$$

Taking into account the Earth radius $r_E = 6378.145\, km$ the orbital radius are:

$$
\begin{aligned}
r_c &= 6776045\, m \\
r_t &= 6778145\, m
\end{aligned}
\tag{5.2}
$$

These information are necessary to calculate the angular velocities

$$
\begin{aligned}
\omega_c &= \sqrt{\frac{\mu}{r_c^3}} \approx 0.001132 \\
\omega_t &= \sqrt{\frac{\mu}{r_t^3}} \approx 0.001131
\end{aligned}
\tag{5.3}
$$

### 5.1.2 Spacecraft Description

In this thesis work, only the chaser characteristics are studied to carry out the rendezvous and docking maneuver.

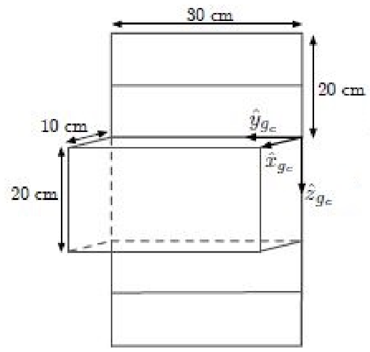


Figure 5.2. Chaser dimensions [3].

The spacecraft taken into account is a 6U CubeSat. The dimension of this spacecraft are $10 \times 20 \times 30\,cm$, as shown in figure 5.2. The wet mass of the entire CubeSat is $m_{c0} = 12\,kg$

Into the simulation, the variable mass is considered. Once the control forces are calculated through the Eq. 2.20, the variable mass is calculated. In fact step by step, the amount of fuel mass consumpted by the actuation system is subtracted in order to simulate as well as possible the mass variation along the entire maneuver.

### 5.1.3 Actuation System

As concern the *Reaction Control System* (RCS), it is composed of four thrusters along each body axis. In [36] the electric propulsion system is analyzed. However, for this thesis work the cold gas propulsion system is taken into account because nowadays it is the most available and most advanced for CubeSats propulsion systems [3]. The misalignment of the thrusters could generate a torque error, but the coupling effect is not considered. The RCS estimated performance are shown in table 5.1.

Table 5.1. RCS performances

| | |
|---|---|
| Thrust | $4\times$ 10 $mN$ per axis |
| Minumum time ON | 25 $ms$ |
| $I_{sp}$ | 60 $s$ |

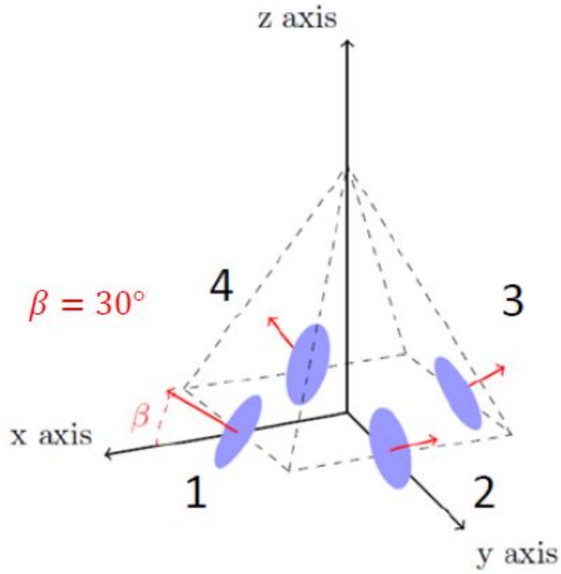


Figure 5.3. RW pyramidal configuration [38].

Table 5.2. RW performances

| | |
|---|---|
| Maximum torque | 1 $Nm$ |
| $I_s$ | 0.3 $kgm^2$ |
| $\alpha$ | 0 $deg$ |
| $\beta$ | 30 $deg$ |

The reaction wheels configuration proposed by [3] is chosen as it provides four wheels in a pyramidal configuration (Figure 5.3), providing redundancy. The datasheet [36] of these COTS wheels describes the performance shown in table 5.2. To transform the controlled torque from body axis to 4 RWs directions, is useful to evaluate the rotation matrix. In fact, starting from the control moment, the three axis moment can be evaluated thanks to the multiplication with the rotation matrix:

$$M_{4RW} = Z^{-1} M_c \rightarrow M_a = Z M_{4RW} \tag{5.4}$$

where

$$Z = \begin{bmatrix} cos(\beta)\cdot cos(\alpha) & -cos(\beta)\cdot sin(\alpha) & -cos(\beta)\cdot cos(\alpha) & cos(\beta)\cdot sin(\alpha) \\ cos(\beta)\cdot sin(\alpha) & cos(\beta)\cdot cos(\alpha) & -cos(\beta)\cdot sin(\alpha) & -cos(\beta)\cdot cos(\alpha) \\ sin(\beta) & sin(\beta) & sin(\beta) & sin(\beta) \end{bmatrix}$$

while $Z^{-1}$ is the *Pseudoinverse Matrix*. As regard $M_c$, this represents the output *Control Moment* given by the Control system. Instead, $M_a$ is the three axis actuation moment calculated with the rotation matrix.

## 5.2 Navigation

In this section, the state observers studied in chapter 3 are applied in the simulation. For each observer, several simulations are performed from different initial conditions of the variables of interest. Starting from zero angular velocity, this value is increased more and more, to assess the quality of the estimation even when in unstable conditions.

In the first part the attitude state variable are estimated, while in the second part the position variable are estimated in order to test the effectiveness of the estimation.

For each observation, the estimation error is calculated as shown in the equation 3.26. At each step $\varepsilon$ is calculated, and then the *root mean square error* rms is computed thanks to *rms* Matlab function. For quaternions, a vector is constructed containing estimation errors for each element, the same thing was done for all angular velocities vector elements. The rms error for each vector is calculated as

$$rms = \sqrt{\frac{1}{N}(x-\hat{x})^2} \tag{5.5}$$

where N is the number of elements of the vector taken into account, necessary to compute the mean value, $x$ is the real value and $\hat{x}$ is the estimated one.

### 5.2.1 Attitude Observation

The initial values of the quaternions and the angular velocities are $q = [0.5,0.5,0.5,0.5]^T$ and $\omega = [0,0,0]^T$.

**Kalman Filter**

The first filter studied for the attitude is the classical *Kalman Filter*. To simulate this observer, a Simulink toolbox is used. The input data needed by this toolbox are: the state space formulation in terms of matrices $A, B, C, D$; the state estimation error covariance $P$; the measurement noise covariance $R$; the process noise covariance $Q$. Moreover the inputs $y$ and $u$ must be defined.

Starting from the inputs $x$ and $u$, they are defined in Eq. 2.10. To better simulate the real scenario, a noise must be added to the process. The noise is simulated with the *Band-limited White Noise* toolbox.

As regard the attitude state variables matrices, they are shown in section 2.2.1. It is important to highlight that these matrices must be discretized, in order to run the simulation. Another important aspect is to define the *step time* $T_s$, which is equal for all the state observers: $T_s = 0.01$ The discretization is computed thanks to a Matlab function written for this purpose.

As regard as the other variables above mentioned, the values chosen for this observer are

Table 5.3. KF parameters for attitude observation

| | |
|---|---|
| Noise Power | 1e-7 |
| $P$ | 1e-6 |
| $Q$ | diag(1e-3 1e-3 1e-3 1e-3 1e-3 1e-3) |
| $R$ | diag(2e-3 2e-3 2e-3 2e-3 2e-3 2e-3) |

The root mean square errors in the estimation with the Kalman filter are:

Table 5.4. RMS error in the estimation of attitude variables with EKF

| | |
|---|---|
| $\varepsilon_q$ [m] | 1.3e-4 |
| $\varepsilon_\omega$ [m] | 9.5e-5 |

### Extended Kalman Filter

As for the Kalman Filter, a Simulink toolbox has been used for the observation with the *Extended Kalman Filter*. As a comparison, the values of $P$, $Q$, $R$ and the noise power are the same of the Kalman Filter. With respect to the KF, the difference is that the inputs of the state variables are not the matrices $A$,$B$,$C$,$D$, but two Matlab functions in which the equation 3.8 are specified. The results in the attitude variables observation are:

Table 5.5. RMS error in the estimation of attitude variables with EKF

| | |
|---|---|
| $\varepsilon_q$ [m] | 1e-5 |
| $\varepsilon_\omega$ [m] | 3.1e-5 |

### 1st order Sliding Mode Observer

As specified in chapter 3, the parameters that must be tuned to obtain a good performance from the 1st order Sliding Mode Observer (SMO) are $\rho$ and $\tau$. The parameters chosen for this observer are

Table 5.6. 1st order SMO parameters for attitude observation

| | |
|---|---|
| $\rho$ | 1e-2 |
| $\tau$ | 1e-1 |

The results in the attitude variables observation are:

Table 5.7. RMS error in the estimation of attitude variables with 1st order SMO

| | |
|---|---|
| $\varepsilon_q$ [m] | 1e-2 |
| $\varepsilon_\omega$ [m] | 1e-3 |

### Super Twisting Sliding Mode Observer

As regard the Super Twisting algorithm, the variables that must be tuned are $\alpha$ and $\beta$ calculated as in Eq. 3.31. These two parameters depend from the variables $a_1$ and $a_2$, chosen as $a_1 = 1.1$ and $a_2 = 1.5$ [24]. In the end, the only one parameter that must be tuned is $C$. The constant $C$ of Eq. 3.31 is 50. Finally, the root mean square estimation error with this observer is:

Table 5.8. RMS error in the estimation of attitude variables with Super Twisting Observer

| | |
|---|---|
| $\varepsilon_q$ [m] | 2.32e-5 |
| $\varepsilon_\omega$ [m] | 2.25e-5 |

As highlighted by the results, starting from the same initial conditions, the worst observer is the $1^{st}$ Sliding Mode Observer. This is a consequence of the system's non-linearity, which does not allow the first order observer to work properly in terms of estimation. As regard as the best observation results, they are given by the Super Twisting Observer that has the lowest RMS error.

To better understand the performance of each observation, several simulations have been tested, starting from different initial conditions.

The following table illustrates the estimation errors for each initial condition.

Table 5.9. RMS error in the estimation of attitude variables starting from different initial conditions

| $q = [0.5,0.5,0.5,0.5]^T$ $\omega = [0,0,0]^T$ | | | | |
|---|---|---|---|---|
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 1.3e-4 | 1e-4 | 1e-2 | 2.32e-5 |
| $\varepsilon_\omega$ [m] | 9.5e-5 | 3.1e-5 | 1e-3 | 2.25e-5 |
| $q = [0.5,0.5,\sqrt{10}/5,\sqrt{10}/10]^T$ $\omega = [0,0,0]^T$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 4e-4 | 1e-4 | 1.2e-2 | 2.36e-5 |
| $\varepsilon_\omega$ [m] | 3.9e-5 | 3.2e-5 | 2.2e-3 | 2.18e-5 |
| $q = [0.5,0.5,\sqrt{10}/5,\sqrt{10}/10]^T$ $\omega = [0.07,0.07,0.07]^T$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 4e-4 | 1e-4 | 1.2e-2 | 2.36e-5 |
| $\varepsilon_\omega$ [m] | 3.9e-5 | 3.2e-5 | 2.2e-3 | 2.51e-5 |
| $q = [\sqrt{5}/5,\sqrt{5}/5,\sqrt{5}/5,\sqrt{10}/5]^T$ $\omega = [0,0,0]^T$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 3.3e-4 | 1e-4 | 1.2e-2 | 2.38e-5 |
| $\varepsilon_\omega$ [m] | 1e-5 | 3.1e-5 | 1e-3 | 1.93e-5 |

Table 5.10. RMS error in the estimation of attitude variables starting from different initial conditions

| | | | | |
|---|---|---|---|---|
| $q = [\sqrt{5}/5, \sqrt{5}/5, \sqrt{5}/5, \sqrt{10}/5]^T$ $\omega = [0.07,0.07,0.07]^T$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 3.4e-4 | 1e-4 | 1.5e-2 | 2.38e-5 |
| $\varepsilon_\omega$ [m] | 1e-5 | 3.1e-5 | 1e-3 | 2.42e-5 |
| $q = [\sqrt{2}/2, \sqrt{10}/10, \sqrt{10}/10, \sqrt{30}/10]^T$ $\omega = [0.1,0.1,0.1]^T$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 9.5e-4 | 1e-4 | 1.1e-2 | 2.39e-5 |
| $\varepsilon_\omega$ [m] | 4.7e-5 | 3.2e-5 | 2.7e-3 | 3.5e-5 |
| $q = [\sqrt{2}/2, \sqrt{10}/10, \sqrt{10}/10, \sqrt{30}/10]^T$ $\omega = [0.5,0.5,0.5]^T$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_q$ [m] | 1.9e-3 | 1e-4 | 1.2e-2 | 2.46e-5 |
| $\varepsilon_\omega$ [m] | 2.8e-3 | 3.1e-5 | 2e-3 | 2.7e-5 |

As expected, by changing the initial conditions a degradation in the observation can be observed. The worst observer is the Sliding mode Observer of the first order. If the step time of the simulation is reduced, there would definitely be a better approximation. However, under the same starting conditions, the first order Sliding Mode Observer reacts worse than the other observers. As for the classical and Extended Kalman filters, their performances remain stable, especially that of the Extended Kalman Filter, which linearizes the equations locally while maintaining good performances. Regarding the Super Twisting Observer, as described in [42], it represents the best choice compared both to the Kalman filter and EKF, since the estimation results are better than the EKF.

### 5.2.2 Position Observation

As for the Attitude, the position and the velocities are estimated along the three axis, starting from initial conditions determined in 5.3. In this case, the y axis dynamics is observed separately from the x and z coupled dynamics, hence two different observations for each state observer are simultaneously computed.

### Kalman Filter

The state variables, just like the attitude, are written in matrix form (2.23 and 2.24) and discretized. The classical Kalman Filter parameters used to estimate x and z position, are shown in the following table:

Table 5.11. KF parameters for x and z position observation

| | |
|---|---|
| Noise Power | 1e-7 |
| $P$ | 1e-6 |
| $Q$ | diag(1e-3 1e-3 2e-3 2e-3) |
| $R$ | diag(1e-2 1e-2 1e-2 1e-2) |

As regard the y axis:

Table 5.12. KF parameters for y axis position observation

| | |
|---|---|
| Noise Power | 1e-7 |
| $P$ | 1e-6 |
| $Q$ | diag(1e-3 2e-3) |
| $R$ | diag(1e-2 1e-2) |

Once the setting parameters are defined and the simulation is completed, the RMS estimation error for each axis is calculated.

Table 5.13. RMS error in the estimation of the position variables with the KF

| | |
|---|---|
| $\varepsilon_x$ [m] | 3.1e-4 |
| $\varepsilon_y$ [m] | 1.2e-4 |
| $\varepsilon_z$ [m] | 3.3e-4 |

### Extended Kalman Filter

As for the attitude, the Extended Kalman Filter is studied starting from the same values of $P$, $Q$, $R$ and noise power of the Kalman Filter. The results in the simulation are showed in table 5.14.

Table 5.14. RMS error in the estimation of position variables with the Extended Kalman Filter

| | |
|---|---|
| $\varepsilon_x$ [m] | 2.1e-4 |
| $\varepsilon_y$ [m] | 3.16e-4 |
| $\varepsilon_z$ [m] | 3.1e-4 |

As it might be expected the results obtained from the Extended Kalman Filter are similar to the classical one. In particular, the performances in the estimation for x and z axis parameters are slightly better, while for the y-axis the result is slightly worse. As from [43], tuning the covariance matrices and the process noise produces better results. The figure 5.4 illustrates the evolution of the estimated

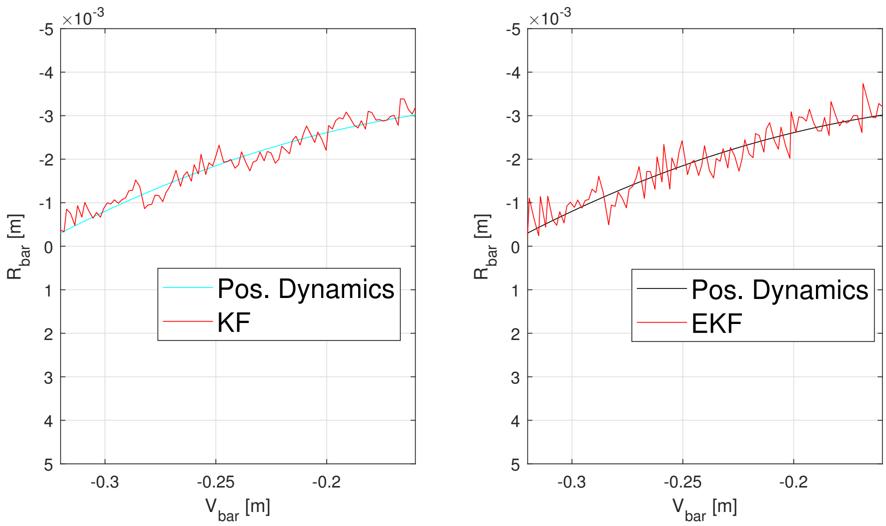


Figure 5.4. Comparison between KF and EKF observation.

variable compared with the position dynamic equations for the Exended and classical Kalman Filter. As it became clear the pattern of the Kalman Filter estimated variable overlaps the real value calculated with 2.19. In this figure a zoom is made so as to understand the real pattern of the estimated value, which furthermore approximates with a small estimation error the real value. By reducing the process noise the result of the estimation is better, $\varepsilon$ decrease and the zigzag motion is reduced, so it is reasonable to think that the pattern obtained is due also to the noise added to the process.

### $1^{st}$ order Sliding Mode Observer

The parameters chosen for this observer have been selected for each axis. The

Table 5.15. $1^{st}$ order SMO parameters for position observation

| | |
|---|---|
| $\rho_x$ | 6 |
| $\rho_y$ | 1e-2 |
| $\rho_z$ | 3 |
| $\tau$ | 1e-2 |

results in the position variables observation are:

Table 5.16. RMS error in the estimation of position variables with $1^{st}$ order SMO

| | |
|---|---|
| $\varepsilon_x$ [m] | 5.8e-2 |
| $\varepsilon_y$ [m] | 4.2e-7 |
| $\varepsilon_z$ [m] | 1.7e-2 |

The results obtained in the position observation for the x and z axis are sightly the same seen in the previous section. It is interesting to notice the good results obtained in y axis estimation.

### Super Twisting Sliding Mode Observer

As regard the Super Twisting algorithm, the variable that must be tuned is $C$, appropriately chosen for each axis.

Table 5.17. Super Twisting Observer parameters for position observation

| | |
|---|---|
| $C_x$ | 6 |
| $C_y$ | 11 |
| $C_z$ | 6 |

In the end, the root mean square estimation errors calculated for this observer are:

Table 5.18. RMS error in the estimation of position variables with Super Twisting Observer

| | |
|---|---|
| $\varepsilon_x$ [m] | 3.1e-4 |
| $\varepsilon_y$ [m] | 9.6e-4 |
| $\varepsilon_z$ [m] | 3.1e-4 |

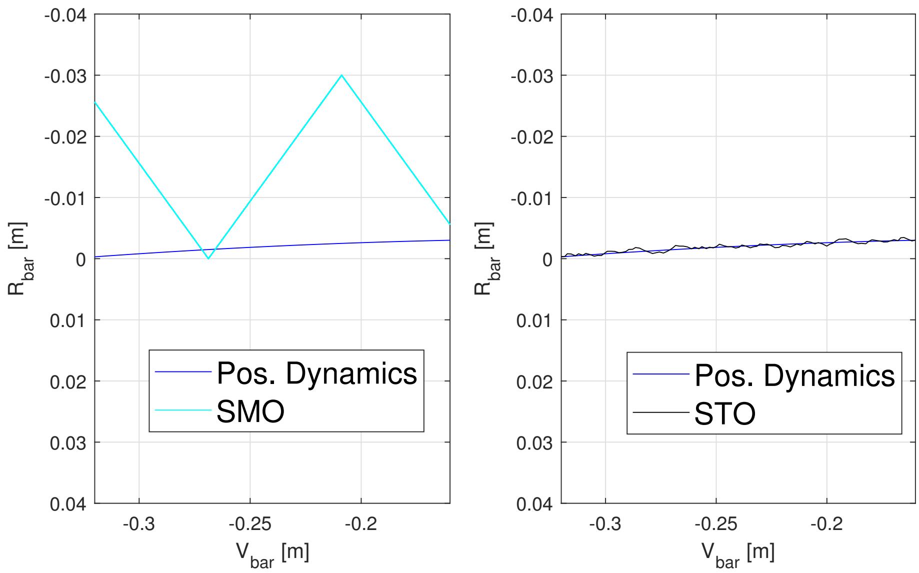


Figure 5.5. Comparison between SMO and STO observation.

As regard as the pattern shown in the left part of the figure 5.5, it illustrates the sign function influence in the calculation of the estimated values. This latter produces the "zigzag" motion around the real value. In the right side of this figure, the pattern of the STO estimated variables points out the influence of the sign function also in this estimation method, which in this way suffers from the chattering behavior.

Analyzing the obtained results, it could be said that Extended Kalman Filter, concerning the position along x and z axis, provides the best results. As for the y axis variable estimation, the first order SMO turns to be the best. The last step is to test, as previously done for the attitude observation, the quality of each observer estimation under different initial conditions. Since the initial position values are bounded to allow the satellite achieving the correct trajectory, only the initial velocities are changed.

Table 5.19. RMS error in the estimation of position variables starting from different initial conditions

| $\dot{x}_0 = 4.15\,m/s, \dot{y}_0 = 0\,m/s, \dot{z}_0 = 0\,m/s$ | | | | |
|---|---|---|---|---|
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_x$ [m] | 3.1e-4 | 2.1e-4 | 5.8e-2 | 3.1e-4 |
| $\varepsilon_y$ [m] | 1.2e-4 | 3.16e-4 | 4.2e-7 | 9.6e-4 |
| $\varepsilon_z$ [m] | 3.3e-4 | 3.1e-4 | 1.7e-2 | 3.1e-4 |
| $\dot{x}_0 = 4.15\,m/s, \dot{y}_0 = 0.5\,m/s, \dot{z}_0 = 0.5\,m/s$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_x$ [m] | 2.1e-4 | 3.1e-4 | 5.2e-2 | 3.11e-4 |
| $\varepsilon_y$ [m] | 1.2e-4 | 1.1e-3 | 5e-2 | 9.6e-4 |
| $\varepsilon_z$ [m] | 8.5e-4 | 3.1e-4 | 1e-2 | 3.11e-3 |

Table 5.20. RMS error in the estimation of attitude variables starting from different initial conditions

| $\dot{x}_0 = 4.15\,m/s, \dot{y}_0 = 2\,m/s, \dot{z}_0 = 1\,m/s$ | | | | |
|---|---|---|---|---|
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_x$ [m] | 2.1e-4 | 3.1e-4 | 5.8e-2 | 3.11e-4 |
| $\varepsilon_y$ [m] | 1.2e-4 | 1.1e-3 | 1.2e-2 | 9.6e-4 |
| $\varepsilon_z$ [m] | 9.8e-4 | 3.1e-4 | 1.8e-2 | 3.12e-3 |
| $\dot{x}_0 = 4.65\,m/s, \dot{y}_0 = 2\,m/s, \dot{z}_0 = 2\,m/s$ | | | | |
| | **KF** | **EKF** | **SMO** | **STO** |
| $\varepsilon_x$ [m] | 2.2e-4 | 3.2e-4 | 3e-2 | 3.11e-4 |
| $\varepsilon_y$ [m] | 1.2e-4 | 1.1e-3 | 1e-2 | 9.6e-4 |
| $\varepsilon_z$ [m] | 1.1e-4 | 3.1e-4 | 2e-2 | 3.11e-4 |

Some interesting consideration can be done about tables 5.19 and 5.20. Starting from y axis position, the results obtained changing initial conditions, show how the first order sliding mode observer deteriorates its performance while increasing the initial velocity. On the contrary, the other filters maintain their performance stable. As regard the x and z axis, the performance of the filters sightly deteriorate although maintaining good results. In particular, the Super Twisting Observer has almost the same performance, and often better, of the other two filters, despite of the different initial conditions. The same results are obtained in [41], where the Extended Kalman Filter is compared to the Sliding mode observer. The main advantage of STO is in its good applicability in most practical cases. In the same way the STO is compared to EKF in [44], with the same results. The robustness of the Super Twisting Observer to noise can be guaranteed because no knowledge of the noise is required. Therefore, it represents the best choice compared to the Kalman Filter and EKF, since the estimation results are comparable to the EKF [42].

## 5.3 Guidance and control

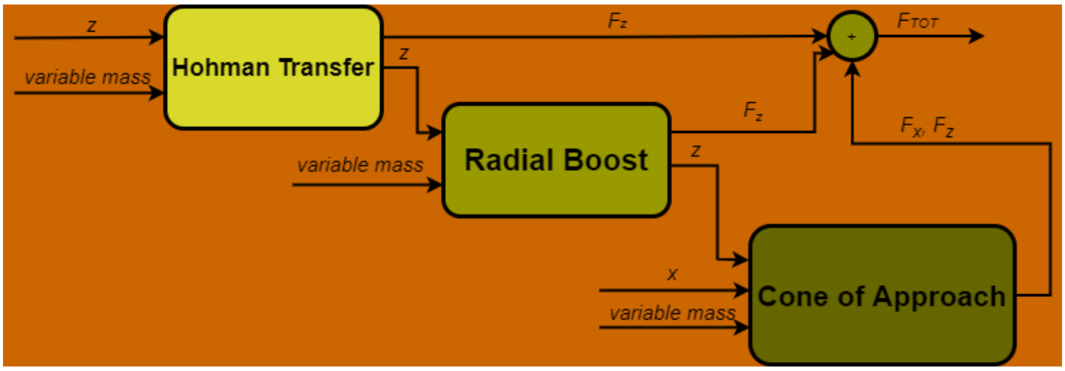


Figure 5.6. Guidance and Control blocks for x and z axis.

The Guidance algorithm is necessary to define the desired trajectory and velocity of the analyzed system. For this simulation, quasi-impulsive maneuvers are applied [22]. The reference mission is shown in figure 2.5 in which the output forces applied in each maneuver are summed to calculate the variable mass, that is used as a input for the maneuvers blocks. In this scheme, all the maneuvers accomplished by the Chaser to reach the Target, shown in Chapter 2, are involved in the guidance block, because each maneuver is subsequent to the previous one.

### Hohmann Transfer

The input data necessary to determinate the $\Delta V_x$ of the Hohmann transfer, $\Delta z = z_t - z_c$ and the Target initial angular velocity $\omega_0$. Another important aspect is to define the initial condition of the maneuver in terms of position and velocity. Into the simulation $\dot{x}_0$ is calculated as:

$$\dot{x}_0 = \frac{3}{2}\omega_0 z_0 + \frac{\omega_0}{4}\Delta z \tag{5.6}$$

Since the final point of the Hohmann transfer is defined by the reference mission ($x_{finalHT} = -500m$ with respect to the Target), the initial position $x_0$ must be calculated as $x_0 = \Delta x + x_{finalHT}$. Finally from 2.1 the $\Delta V_x$ can be calculated. The results are shown in the following table:

Table 5.21. Hohmann Transfer parameters

| | | | |
|---|---|---|---|
| $\Delta z$ | 2100 $m$ | $\Delta x$ | 4950 $m$ |
| $\Delta V_x$ | 0.5942 $m/s$ | | |
| $x_0$ | -5.450 $m$ | $y_0, z_0$ | 0 $m$ |
| $\dot{x}_0$ | 4.15 $m/s$ | $\dot{y}_0, \dot{z}_0$ | $0m/s$ |

### Radial Boost

To reach the final position $S2_4$ four different radial boost maneuvers are needed. The input data to calculate the $\Delta V_z$ are the $\Delta x$ and $\omega_0$, as shown in equation 2.2. The values for each radial maneuver are listed in the following table:

Table 5.22. Radial boost parameters

| | | | |
|---|---|---|---|
| $\Delta x_1$ | 200 $m$ | $\Delta x_3$ | 100 $m$ |
| $\Delta Vz_1$ | 0.0733 $m/s$ | $\Delta Vz_3$ | 0.0461 $m/s$ |
| $\Delta x_2$ | 150 $m$ | $\Delta x_4$ | 35 $m$ |
| $\Delta Vz_2$ | 0.0478 $m/s$ | $\Delta Vz_4$ | 0.0147 $m/s$ |

### Cone of Approach maneuver

In the final corridor, the satellite must be enclosed in a cone for security reasons. The value of the half cone angle is calculated as follow:

$$\theta = arctg(\frac{r_1 - r_2}{d_1}) \tag{5.7}$$

where $r_1 = 7m$ is the initial point maximum height and $r_2 = 0.1m$ is the final point maximum height, next to the docking port. $d_1 = 15m$ is the relative distance between the Chaser and the Target. With these values $\theta = 0.4311rad$.

The impulse is a straight line $V_{bar}$ approach with constant velocity. It requires a continuous application of thrust to counteract the Coriolis forces. The values of the $\Delta V_x$ chosen is: $\Delta V_x = 0.15\, m/s$.

In this phase, z axis must be controlled to ensure the Chaser inside the security cone. To control this variable, a PID controller is necessary. The PID gains chosen are:

Table 5.23. PID parameters for z axis control

| | |
|---|---|
| $K_P$ | 20 |
| $K_I$ | 1 |
| $K_D$ | 300 |

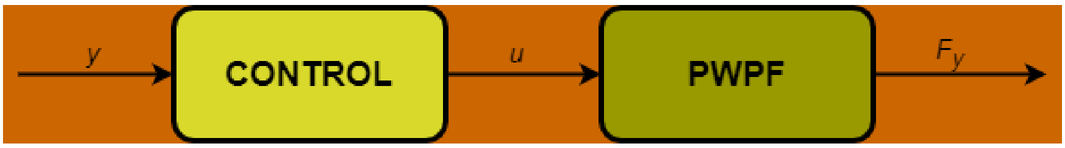


Figure 5.7. Guidance and Control blocks for y axis.

As concern the position dynamics along the y axis, it is not influenced from the other maneuver, because as represented in 2.19, the y dynamics is decoupled from x and z. The aim is to keep this variable around zero. To achieve this goal both a *Sliding Mode Control* (SMC) and a PID controller are designed and tuned in order to define the best choice for this variable control. The PID parameters chosen are

Table 5.24. PID parameters for y axis control

| | |
|---|---|
| $K_P$ | 5 |
| $K_I$ | 1 |
| $K_D$ | 8 |

while the SMC parameters are:

Table 5.25. SMC parameters for y axis control

| | |
|---|---|
| $c$ | 1.5 |
| $\rho$ | 2 |

As concern the external disturbances considered for the position evolution, the air drag model and a constant value, that considers other external disturbances, are taken into account for the simulation. From 2.26

$$F_D \approx 4 \times 10^{-5} N \tag{5.8}$$

while the constant force is assumed as:

$$F_{ext} = 1 \times 10^{-5} N \tag{5.9}$$

As regard the disturbances affecting the y axis, a constant disturbance of $0.001N$ is considered.

Finally in the following figures,the trajectory is analyzed alogn the three axis and considering the x-z plane. As evident the control of each variable is computed, and the pattern of the figures follows the reference mission 2.5.

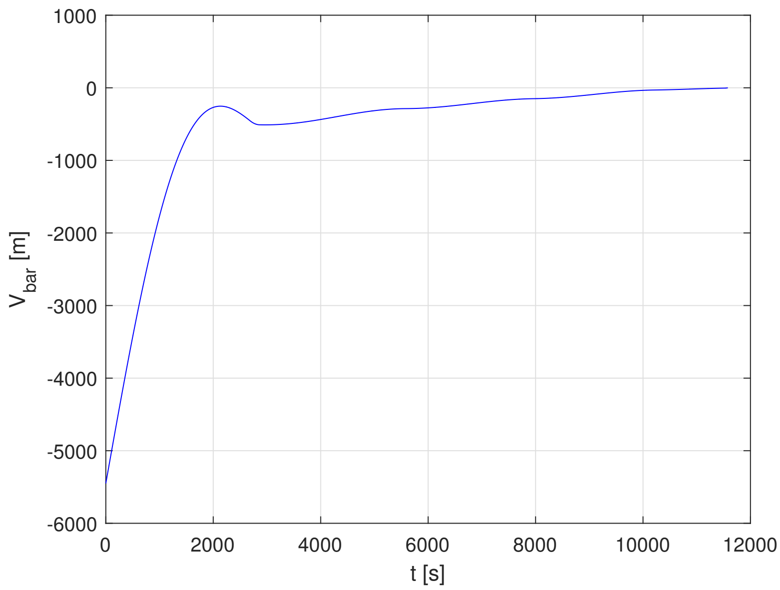


Figure 5.8. Evolution of x position.

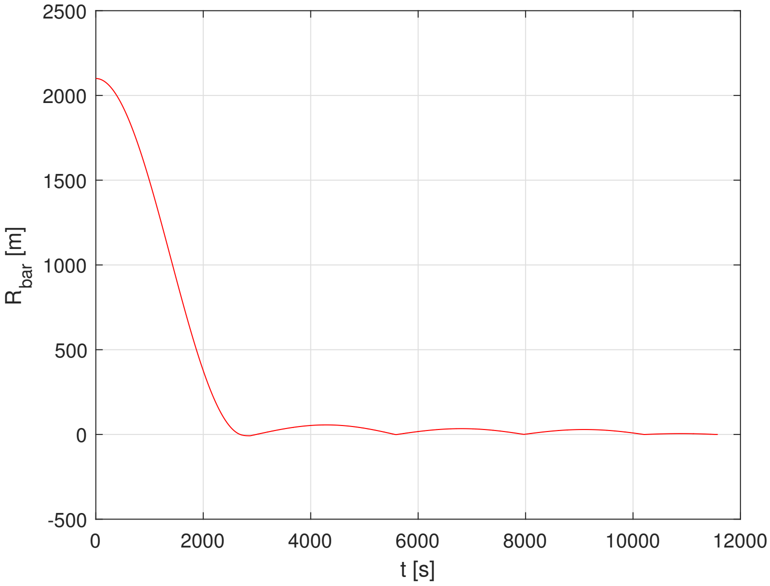


Figure 5.9. Evolution of z position.

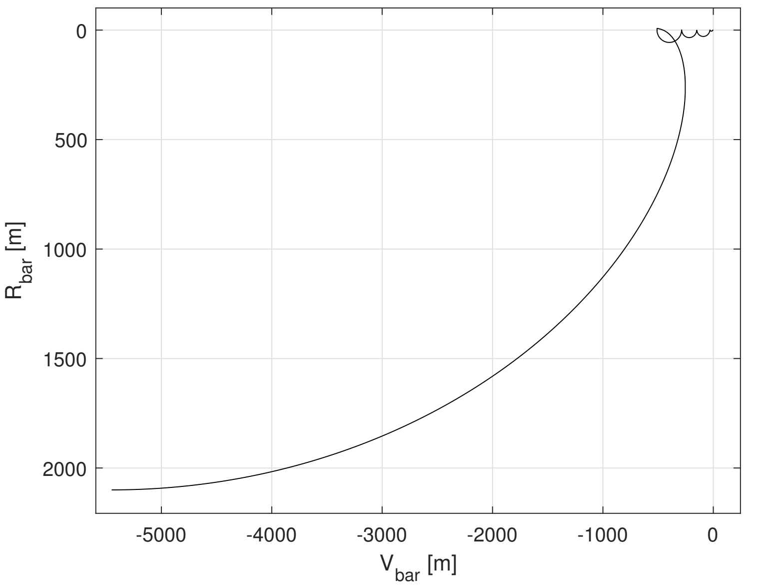


Figure 5.10. Complete manoeuvre in the x-z plan.

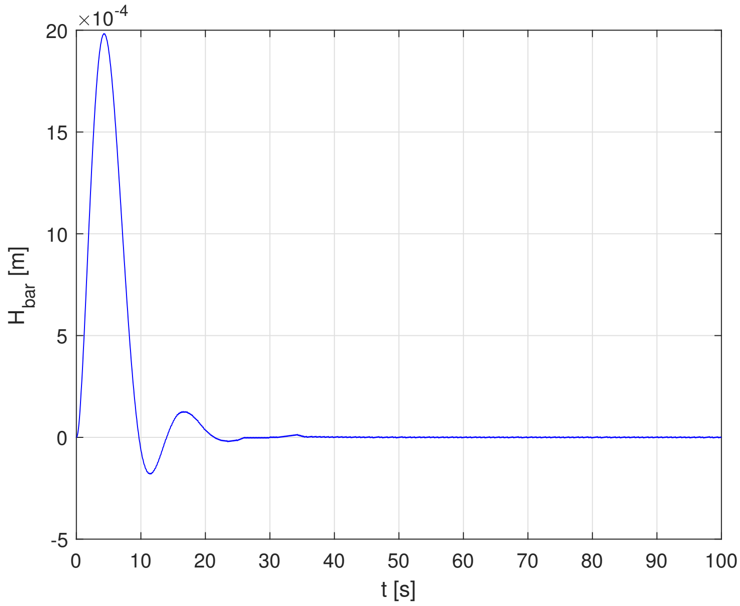


Figure 5.11. Evolution of y position with PID control.

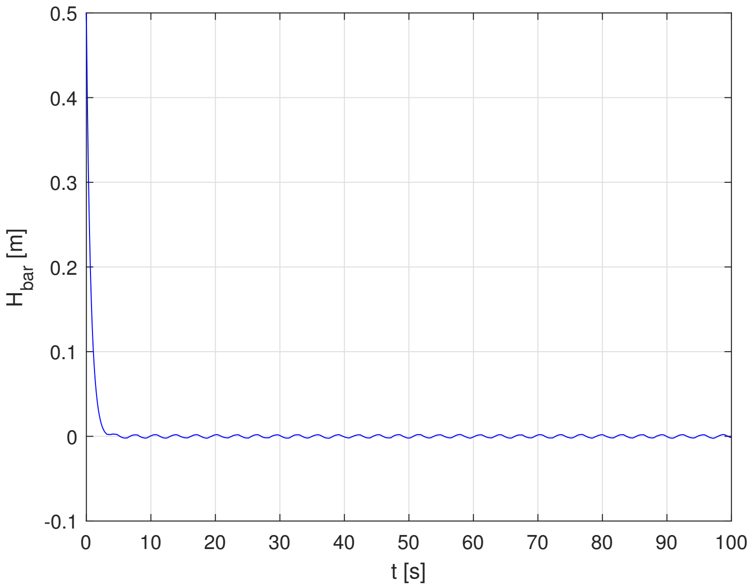


Figure 5.12. Evolution of y position with SMC control.

As concerning the y position control, the PID control is chosen. As it can be seen in figure 5.12, the sign function influence the motion of this variable. To solve this problem, the sigmoid function could be used, as explained in (3.22).

## 5.4 Attitude Control

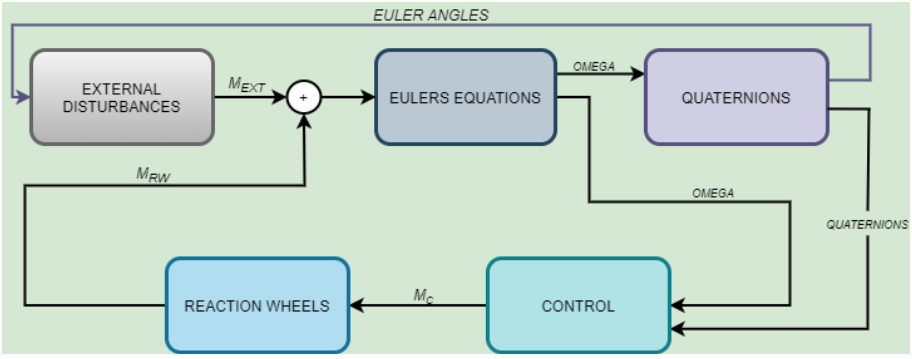


Figure 5.13. Attitude determination and control block diagram.

To determinate the attitude variables, the mathematics model used for the determination of each variables are shown in section 2.2.1. Considering the external disturbance model, the gravity gradient model is implemented. From Eq. 2.27 the Gravity Gradient torque depends from the Euler angles, thus the gravity gradient torque is defined as:

$$T \approx 3n^2 \cdot \begin{bmatrix} (I_z - I_y)\phi \\ (I_z - I_x)\theta \\ 0 \end{bmatrix} \tag{5.10}$$

$n = \sqrt{\mu/a^3}$ is the orbital rate, where $a$ is the semi-mayor axis of the orbit. $\phi$ and $\theta$ are respectively the roll and pitch Euler angle. In addiction to the gravity gradient torque, a constant disturbance torque is added to simulate the other external disturbances shown in section 2.3. The constant value is

$$M_{ext} = 0.001Nm \tag{5.11}$$

The reaction wheels model is explained in 5.1.3 and the values that must be tuned for the simulations are the transfers functions coefficients. As concern the numerator coefficient $K_r$ and the denominator coefficient $\tau$ the following values are chosen: $K_r = 1$ and $\tau = 1$.

Finally, as regard as the control block, different controllers have been used for the attitude. The first one is the *Quaternion feedback controller* (QFC), explained in chapter 4. This controller is used as a comparison with the Linear Quadratic

Gaussian. In the end, the combination between the LQR controller and different state observers is tested.

Starting from the QFC, as shown in 4.13 and 4.14, two different approaches in the definition of the tuning parameter $K_p$ can be applied. For this thesis work, the second approach is applied so the parameters $\alpha$ and $\beta$ are defined as

$$\alpha = 0.01$$

$$\beta = 0.02$$

As regard as $K_d = diag(d_1, d_2, d_3)$, it is defined as follow

$$K_d = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 8 \end{bmatrix} \tag{5.12}$$

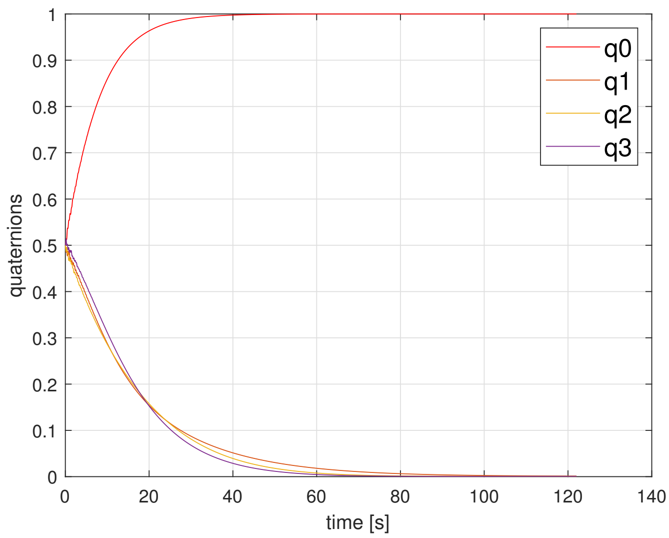


Figure 5.14. Variation of quaternion with QFC Control.

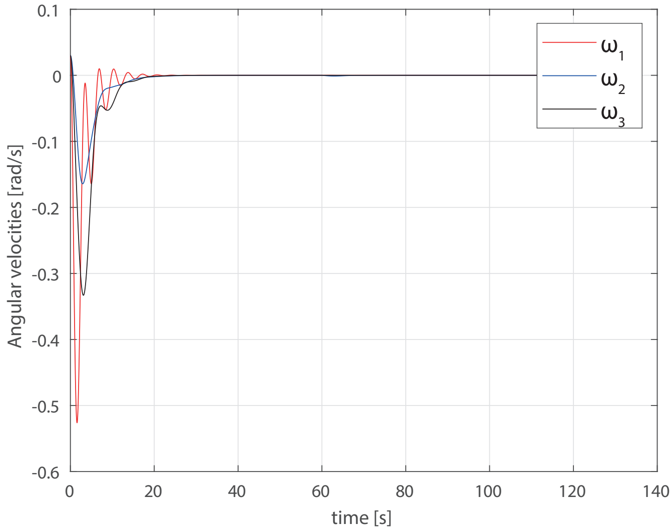


Figure 5.15. Variation of angular velocities with QFC Control.

The result in the control with the QFC are shown in figures 5.14 and 5.15. Both the quaternions and the angular velocities reach the desired value in about 60 seconds for the quaternions and 20 seconds for the angular velocities, with an initial oscillation.

As regard as the application of the Linear Quadratic Gaussian, the LQR control gain must be calculated. First the weights $Q$ and $R$ must be defined:

$$Q = \begin{bmatrix} 15e-4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15e-4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 15e-4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15e-4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15e-4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15e-4 \end{bmatrix} \tag{5.13}$$

$$R = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix} \tag{5.14}$$

The LQR gain $K_r$ is then calculated starting from the Riccati equation, with the following results:

$$\begin{bmatrix} 0.0387 & -2.6e-6 & 9.1e-5 & 0.0552 & 6.1e-6 & 1.6e-5 \\ 0 & 0.0387 & -8.3e-10 & 6.8e-7 & 0.1243 & -2.7e-9 \\ -1.2e-4 & 7e-10 & 0.0387 & 3.9e-6 & -6.2e-9 & 0.0877 \end{bmatrix} \tag{5.15}$$

The results of this control design are shown in the following figures.

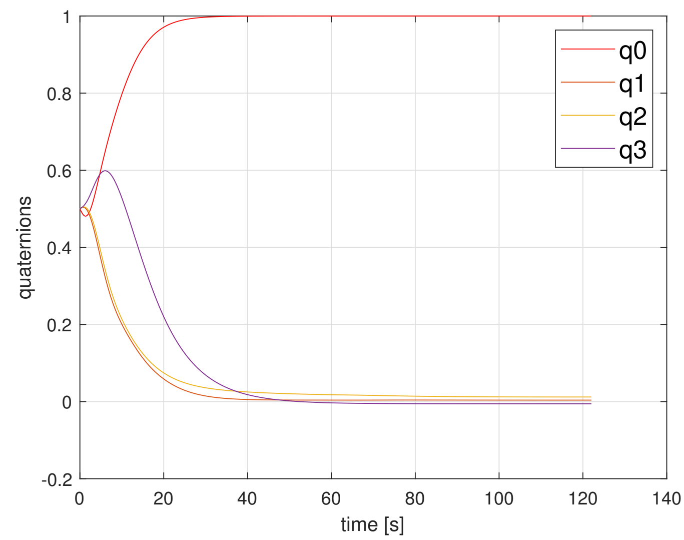


Figure 5.16. Variation of quaternion with LQG Control.

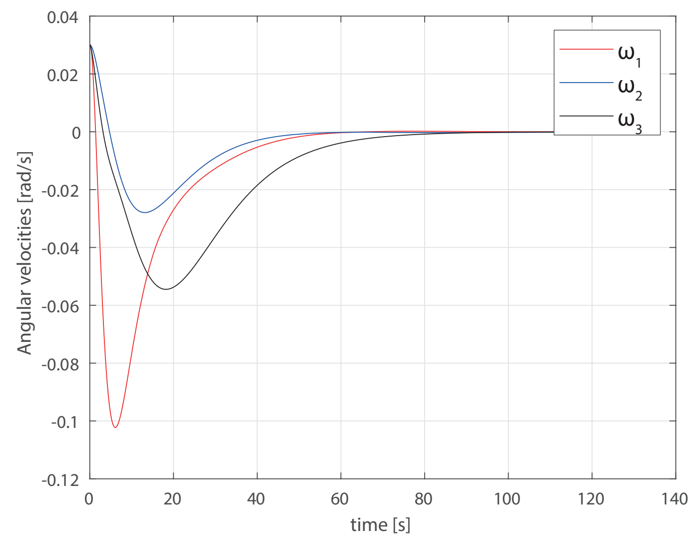


Figure 5.17. Variation of angular velocities with LQG Control.

As from Figures 5.16 and 5.17, the LQG control works better with respect to the QFC in term of stability of the attitude variables. Indeed, if with the QFC control an initial oscillation of the angular velocities along the three axis happens, in this kind of control the rotation immediately dumps. The settling time is approximately the same for the quaternions, while it is higher for the angular velocities.

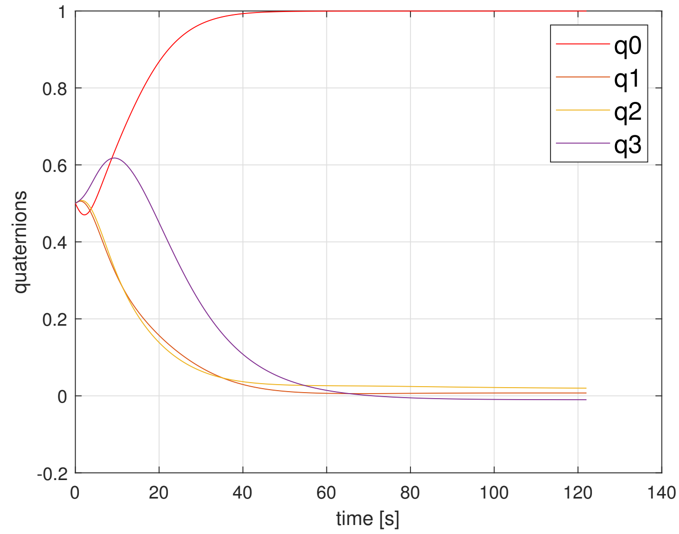


Figure 5.18. Variation of quaternion with LQR + EKF Control.

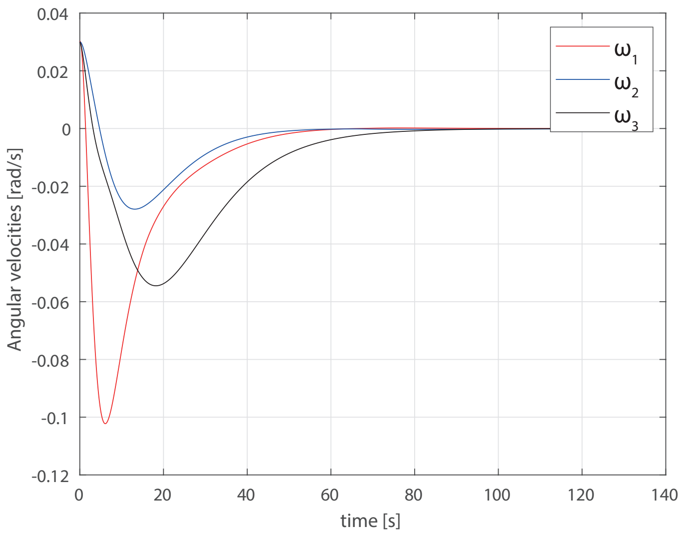


Figure 5.19. Variation of angular velocities with LQR + EKF Control.

In the end, the LQR optimal control is tested in combination with the Extended Kalman Filter and the Super Twisting Observer. The controller works thanks to the separation principle explained in section 4.2.2. Approximately the same result of the LQG control is obtained, considering the same LQR gains for each controller. Obviously, more the estimation is accurate, more the control will be accurate.

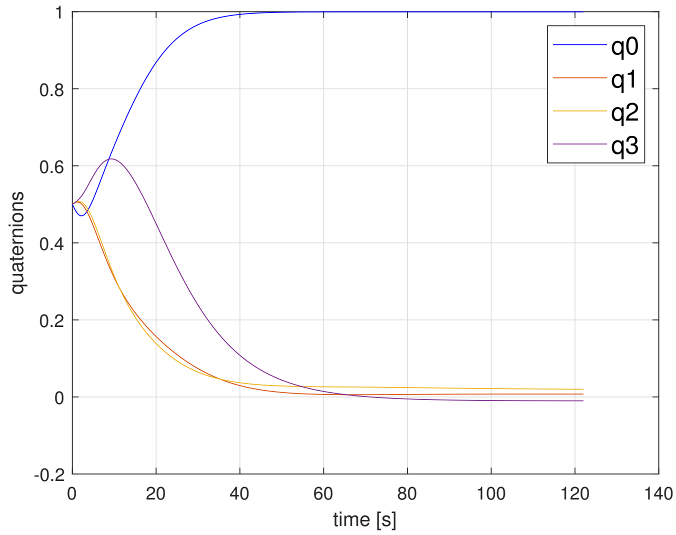


Figure 5.20. Variation of quaternion with LQR + STO Control.

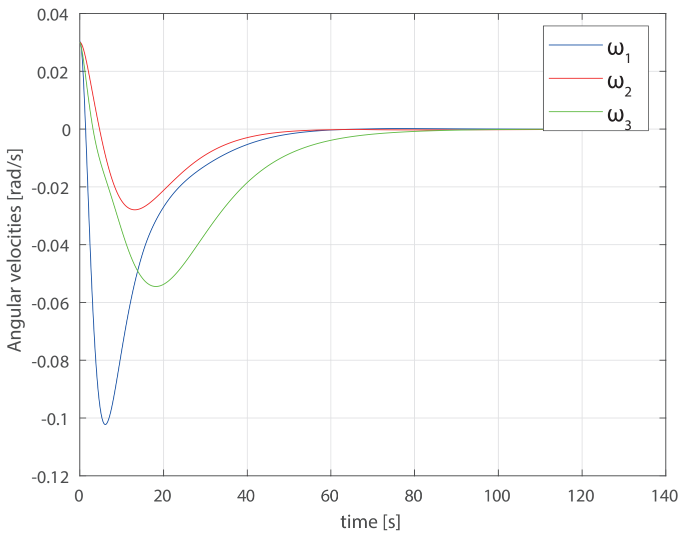


Figure 5.21. Variation of angular velocities with LQR + STO Control.

# Chapter 6

# Conclusions and Future Works

In this thesis, a comparison of four different navigation algorithms is proposed: Kalman Filter, Extended Kalman Filter, Sliding mode Observer and Super Twisting Observer. These filters are involved in a rendezvous and docking simulation with the aim of testing the effectiveness of the estimation.

A further objective is the control of all the variables, allowing the Chaser reaching the Target. Four different controllers are implemented: the classical PID, the Sliding mode Controller, the Quaternion Feedback Controller and the Linear Quadratic Gaussian. The latter is a combination of both the optimal LQR control and the Kalman Filter. The LQG is tested only for attitude control and he results obtained through the LQG are better in terms of control stability than the QFC. In the same way, the LQR combined with the STO and EKF provides good results. This means that a good estimate can provide also a good control, if the *separation principle* can be applied. Therefore, this thesis offers new insights analysis for further research, through the control properties for unstable and highly non-linear systems.

As concerning the navigation algorithm, the main idea is to estimate all the state variables involved in the simulation. The goal is to keep the estimated values close to the real one, calculated through the dynamic equations. Once the observation parameters have been defined and set, the simulation is done starting from the same initial conditions. Subsequently, these latter are changed to test their observation response.

By changing the initial conditions, each observer maintains its performance other than the $1^{st}$ Sliding Mode Observer. Therefore, as the results given by the latter prove, the SMO performance are not comparable to the other three observers, which maintain higher performances. A further step could definitively be to tune the parameters, while reduce the step time of the simulation, so to obtain a better estimation.

Thereafter, a comparison between the Kalman Filter and the Extended Kalman Filter takes place, setting the same parameters. In line with the results obtained in [42], the EKF produces the best estimation results, although depending on appropriate selection of the measurement and process noise matrices. Moreover, the EKF maintains its performance despite the initial condition change. If Q and R matrices are well tuned, EKF stability and convergence reveal to be good.

Finally, the Super Twisting Observer is studied by comparing the results with the well known Kalman filters. As proved in [43], it is much simpler to implement this observer, given that the dynamic performance can be modified although maintaining good results. No knowledge of noise statistic is required, hence the Super Twisting Observer turns out to be the best choice compared to the Kalman filter in terms of robustness to external noise. Despite the suffering caused by the chattering behavior, the STO steers satisfactorily the system, often obtaining better estimation results than the EKF.

The following steps could be to simulate the maneuver in presence of bigger external disturbances and unexpected changes in the dynamics, trying to understand the response of the filter in terms of estimation errors and observation stability. This could represents a significant breakthrough for the design of sensor-less systems.

# Bibliography


[1] Y. Shtessel,C. Edwards,L. Fridman,A. Levant, *"Sliding Mode Control and Observation"*, New York: springer New Nork, 2014

[2] A. S. I. Zinober, *"Deterministic control of uncertain systems"* Proceedings. ICCON IEEE International Conference on Control and Applications, Jerusalem, Israel, 1989, pp. 645-650, doi: 10.1109/ICCON.1989.770598.

[3] C.S. Pirat, *"Guidance, Navigation and Control for Autonomous Rendezvous and Docking of Nano-Satellites"*, Lausanne, EPFL, 2018

[4] E. Lavretsky, *"Adaptive control: Introduction, overview, and applications"*, in Lecture notes from IEEE Robust and Adaptive Control Workshop, 2008.

[5] Wikipedia.en, *"Adaptive Control"*,$https : //en.wikipedia.org/wiki/Adaptive_control$,2020

[6] Z. Öreg, H. Shin and A. Tsourdos, *"Model identification adaptive control - implementation case studies for a high manoeuvrability aircraft"* ,2019 27th Mediterranean Conference on Control and Automation (MED), Akko, Israel, 2019, pp. 559-564, doi: 10.1109/MED.2019.8798513.

[7] V. Van Doren, *"Fundamentals of Self-Tuning Control"*, Control Engineering, Vol. 54, No. 7, July, 2007.

[8] I. Ullah, X. Su, J. Zhu, X. Zhang, D. Choi, Z. Hou, *"Evaluation of Localization by Extended Kalman Filter, Unscented Kalman Filter, and Particle Filter-Based Techniques"* , Wireless Communications and Mobile Computing, vol. 2020, Article ID 8898672, 15 pages, 2020. https://doi.org/10.1155/2020/8898672 Hou;

[9] I. Ullah, X. Su, X. Zhang, D. Choi, *"Simultaneous Localization and Mapping Based on Kalman Filter and Extended Kalman Filter"*, Wireless Communications and Mobile Computing, vol. 2020, Article ID 2138643, 12 pages, 2020. https://doi.org/10.1155/2020/2138643

[10] Mathworks.com - *"Understanding Kalman Filters - part 1"*, $https : //bit.ly/39xj8Aa$, 2017

[11] Mathworks.com - *"Understanding Kalman Filters - part 2"*, $https : //bit.ly/3rEIpOO$, 2017

[12] Mathworks.com - *"Understanding Kalman Filters - part 3"*, $https : //bit.ly/2QVO7zl$, 2017

[13] Notes course on Advanced Sliding Mode Control and Estimation for Real Complex Systems, *"Introduction to Sliding Mode Observers I"*, university of IIT

Roorkee, Prof. S.K. Spurgeon, 2017;

[14] F. Okoli,*"Kalman Filter Modelling and Simulation"*, 2015

[15] H. Leeghim, H. Bang, Y. Choi, *"Adaptive attitude control of spacecraft using neural networks"*,Acta Astronautica,Volume 64, Issues 7–8, 2009

[16] M. Mammarella, D. Y. Lee, H. Park, E. Capello, M. Dentis, G. Guglieri, *"Attitude Control of a Small Spacecraft via Tube-Based Model Predictive Control"*, Journal of Spacecraft and Rockets. 56. 1-18. 10.2514/1.A34394, 2019

[17] A. Shekhar and A. Sharma,*"Review of Model Reference Adaptive Control"* International Conference on Information , Communication, Engineering and Technology (ICICET), Pune, India, 2018, pp. 1-5, doi: 10.1109/ICICET.2018.8533713.

[18] V. Vandoren, *"Auto-Tuning Control Using Ziegler-Nichols"*, PH.D., Control engineering, October 1, 2006

[19] S. L. Brunton, J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge: Cambridge University Press. doi:10.1017/9781108380690, 2019

[20] A. Chalanga, S. Kamal, L. M. Fridman, B. Bandyopadhyay and J. A. Moreno,*"Implementation of Super-Twisting Control: Super-Twisting and Higher Order Sliding-Mode Observer-Based Approaches"* in IEEE Transactions on Industrial Electronics, vol. 63, no. 6, pp. 3677-3685, June 2016, doi: 10.1109/TIE.2016.2523913.

[21] Y. Yang, *"Controllability of Spacecraft Using Only Magnetic Torques"*,IEEE Transactions on Aerospace and Electronic Systems pp.955-963, VOL. 52, NO. 2 April 2016

[22] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*, (Cambridge Aerospace Series). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511543388, (2003)

[23] A. Becker, *"KalmanFilter.net"*,$https://www.kalmanfilter.net/default.aspx$, 2018

[24] J. Davila, L. Fridman and A. Levant,*"Second-order sliding-mode observer for mechanical systems"* in IEEE Transactions on Automatic Control, vol. 50, no. 11, pp. 1785-1789, doi: 10.1109/TAC.2005.858636, Nov. 2005

[25] A. E. Golubev, A.P. Krishchenko, S.B. Tkanchev, *"Separation principle for a class of nonlinear systems"*,15th Triennial World Congress, Barcelona, Spain, 2002

[26] I. Salgado, I. Chairez., J. Moreno, L. Fridman, A. Poznyak, *"Generalized Super-Twisting Observer for Nonlinear Systems"*, IFAC Proceedings Volumes, Volume 44, Issue 1, 2011

[27] L. Vile, *"Sliding Mode Control Toolbox"* , $https://www.mathworks.com/matlabcentral/fileexchange/71174-sliding-mode-control-toolbox$, MATLAB Central File Exchange. Retrieved March 30, 2021.

[28] Y. A. Khan "Sliding Mode Control Lectures", $https://bit.ly/3deYp54$, 2018

[29] H. Bang,C.K. Ha, J. H. Kim, *"Flexible spacecraft attitude maneuver by application of sliding mode control"* ,Acta Astronautica,mVolume 57, Issue 11, 2005

[30] A.B. Younes, J. D. Turner, D. Mortari, J. Junkins , *"A Survey of Attitude Error Representations"*, In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, USA, 13–16 August, 2012

[31] G.C. Goodwin, S.F. Graebe, M.E. Salgado, *"Control systems design"*,Int. J. Adapt. Control Signal Process. 16, 173–174, 2002

[32] J. W. Webb, R. A. Reis, *"PID Control of Continuous Processes"*,$https://www.ni.com/it-it/innovations/white-papers/06/pid-theory-explained.html$, 2020

[33] Wikipedia.en, *"PID controller"*, $https://en.wikipedia.org/wiki/PID_controller$, 2021

[34] MathWorks.com, *"Linear-Quadratic-Gaussian (LQG) Design"*, $https://it.mathworks.com/help/control/getstart/linear-quadratic-gaussian-lqg-design.html$, 2018

[35] S. Brunton, *"Control Bootcamp: Linear Quadratic Gaussian (LQG)"*, $https://bit.ly/3frKlIb$, 2017

[36] J.B. Mirczak, *"Milli-Newton Thrust Stand For Electric Propulsion"*,Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, 2003

[37] Hyperion Technologies, *"RW400"*,$http://hyperiontechnologies.nl/products/rw400-series-reaction-wheel/$ 2021

[38] Notes Course Dynamics and Control of Space Vehicles, *"Quaternion feedback control with RCS and RW"*, Polytechnic of Turin, Prof. E. Capello, 2019;

[39] Notes Course Dynamics and Control of Space Vehicles, *"Position and attitude dynamics"*, Polytechnic of Turin, Prof. E. Capello, 2019;

[40] Notes Course Dynamics and Control of Space Vehicles, *"External"*, Polytechnic of Turin, Prof. E. Capello, 2019;

[41] Y. Zhang, Z. Zhao, T. Lu, L. Yuan, W. Xu and J. Zhu, *"A comparative study of Luenberger observer, sliding mode observer and extended Kalman filter for sensorless vector control of induction motor drives"*, IEEE Energy Conversion Congress and Exposition, San Jose, CA, USA, 2009, pp. 2466-2473, doi: 10.1109/ECCE.2009.5316508, 2009

[42] F. Chen, M.W. Dunnigan, *"Comparative study of a sliding-mode observer and Kalman filters for full state estimation in an induction machine"*, Electric Power Applications, IEE Proceedings -. 149. 53 - 64. 10.1049/ip-epa:20020018, 2002

[43] H. Majid, H. Abouaissa, *"Comparative Study of a Super-Twisting Sliding Mode Observer and an Extended Kalman Filter for a Freeway Traffic System"*, Published Online 03 Jul 2015

[44] J. Qi, A. F. Taha and J. Wang, *"Comparing Kalman Filters and Observers for*

*Power System Dynamic State Estimation With Model Uncertainty and Malicious Cyber Attacks"*, in IEEE Access, vol. 6, pp. 77155-77168, doi: 10.1109/ACCESS.2018.2876883, 2018